The Dissertation Committee for Kranthi Mitra Adusumilli
certifies that this is the approved version of the following dissertation:

Scheduling and Control of Stochastic Processing Networks

Committee:

John Hasenbein, Supervisor

Gustavo de Veciana

David Morton

Elmira Popova

Erhan Kutanoglu

Scheduling and Control of Stochastic Processing Networks

by

Kranthi Mitra Adusumilli, BTech, MSE

Dissertation

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Doctor of Philosophy

The University of Texas at Austin

August 2007

Dedicated to my parents, grandparents and sister

Acknowledgments

I am greatly indebted to numerous people including faculty members, family members and friends. Without their continuous support and encouragement this dissertation would not have been possible. I hope that this work does justice to the support they have given me.

Firstly, I express my sincere gratitude to my advisor Dr. John Hasenbein for always being there, his patience and his encouraging words. I have benefited greatly under his wonderful mentorship, and the freedom given in research. The numerous insightful discussions have helped me develop my understanding of various topics.

I would like to thank the committee members Dr. Morton, Dr. Popova, Dr. Kutanoglu and Dr. Veciana for their support and valuable suggestions. Their support and approval has meant a lot to me. I will always be grateful to the various faculty members in The University of Texas at Austin who have taught me and have been pivotal during my doctoral studies.

The confidence, understanding and compassion my parents and sister have shown towards me, especially during my graduate studies has been a constant source of motivation. It goes without saying that I am deeply indebted to them.

I would not have explored the territory of doctoral studies if not for the encouragement from Nagender, Kranthi Reddy, Silpa and Anand. I thank them for this. During the initial drafts, numerous suggestions from Srikanth and Priti have been very helpful and made the later revisions easier. I thank them for having to put up with me and their support. I am grateful to Anubhav and Mohit for their understanding and help during the initial critical phase of graduate life. I express my gratitude to Amit, Burak and other past and present ORIE students for numerous discussions and sharing their knowledge. The doctoral studies would have not have been such pleasant and memorable journey if it was not for Arvind, Ankur, Aditya, Abdul, Asu, Babaji, Bharani, Deepti, Divi, Eashwar, Naveen, Sandeep, Sameer, Sandhya, Yak, Ashwin and numerous others. I thank them for their support. I sincerely apologize if I have failed to mention someone I ought to.

I would like to thank International SEMATECH for their continued finance assistance during my graduate studies. I am especially grateful to Robert Wright. I take this opportunity to thank Dr.Harrison for suggesting the single server control problem and Dr. Basamboo for providing insight into the problem.

Scheduling and Control of Stochastic Processing Networks

Publication No. ____________________

Kranthi Mitra Adusumilli, Ph.D.
The University of Texas at Austin, 2007

Supervisor: John Hasenbein

In this dissertation the following two control problems of different queueing systems are addressed.

Service rate and Admission Control: We consider a single server system with constant Poisson arrivals subjected to both service rate and admission controls. The controller can admit or reject a customer on arrival and choose the service rate, from a fixed subset, when an arrival or departure occurs. With each control decision is associated a one time rejection cost and a cost for service. A holding cost and cost for service are continuously incurred.

The holding cost is non-decreasing in the number of customers in the system and the cost for service is non-decreasing in the service rate. The objective is to minimize the long run average cost per unit time. We restrict to (state-dependent) stationary deterministic controls and derive the optimality equations. We use standard average cost dynamic programming techniques to obtain the optimality equations in terms of minimum achievable average cost. The stationary controls correspond to a threshold system state (finite or infinite) and service rates for each of the states. Customer are rejected service once the number of customers in the system is greater than or equal to the threshold. We suggest a fast scheme, based on considering incremental values of the system threshold for computing the optimal average cost and the associated optimal service rates. This is similar to initially fixing a system threshold, and choosing the optimal service rates thereafter. We establish the monotonicity of optimal service rates in terms of the queue lengths, for the original system as well as the intermediate systems. Finally, we prove that the constructed stationary optimal policy is optimal across all possible non-anticipative controls.

Stochastic Scheduling under Parameter Uncertainty: We suggest a new approach to model randomness in the context of job shop scheduling. In addition to inherent randomness such as variable processing times for a job class, certain parameters, e.g. like the initial number of jobs might not be known with certainty. We consider scheduling of such a system: a stochastic job shop with parameter uncertainty.

We model a situation in which the initial number of jobs and the mean processing times of jobs are uncertain. We assume that the controller has a limited ability to make certain control decisions before the initial number of jobs and processing times are revealed. Thus these decisions must be made a priori. For each server, the scheduler must choose a cycle length and the fraction of time devoted to processing each job class during a cycle. Under these assumptions the resulting optimization problem for minimizing expected makespan is a stochastic integer program. We obtain a *continuous job shop model*, a relaxation of the original model, by relaxing the integrality requirements. When we restrict allowable policies to satisfy certain time allocation constraints, we obtain a further relaxation called the *fluid model*. The optimal solution for the fluid model is shown to be optimal for the continuous job shop. The optimization problem for the fluid model is a stochastic non-linear program, which is easier to solve. Based on the optimal solution to the fluid model, we propose a scheduling heuristic. We show the asymptotic optimality of the heuristic. The optimality results is in terms the expected makespan. We also extend the asymptotic optimality results to the case when processing times are random, i.e., when the job completion process is doubly stochastic. We obtain tight bounds for makespan which hold with high probability. We also discuss an assignment problem for the single station case and suggest asymptotically optimal heuristics, which are constructed from the associated fluid model.

Contents

Chapter 1

Introduction

Modeling and analyzing a complex real life system in its completeness can be a difficult, time consuming and overwhelming task. Simple stylized models promise to provide important insights that can be good guidelines for controlling the actual complex systems. With the goal of gaining insight into complex processing networks, we analyze two different types of stylized queueing systems. These models are motivated by practical problems. As is the case for any system, it is desirable to manage these systems so as to achieve certain desirable characteristics such as small queues. We try to provide guidelines for optimal or near optimal control for the stylized queueing systems for simple objective functions. Below is a brief description of these models and their features.

The first queueing model is a single server system with both service rate and admission controls. In a manufacturing center, the manager might have the choice to change the processing capability dynamically or outsource the order at a fixed cost to an alternate contractor. In light of this simple example, we consider an idealized queueing model, where the system manager can accept or reject arriving customers. Furthermore, she can change the rate at which the customer is served. Building on the work of George and Harrison [13], this dissertation expands on the original problem by incorporating admission control. We assume mild conditions, which allow the inclusion of features like unbounded service rates and non-convex cost functions. The problem is a Markov decision process with an objective of minimizing the long run average cost of the system. A main contribution of this analysis is proving the existence of stationary optimal control among the class of all possible non-anticipative policies. The methodology suggested to compute optimal or near-optimal stationary control is another notable contribution. We develop a numerical method which provides controls which converge to the optimal control policy.

The second model is a stochastic job shop with uncertain parameters. Firstly, we present a motivating example. In the case of a natural calamity, as a disaster management measure, traffic lights in the city should be synchronized to evacuate the city quickly. However, the number of people taking a specific route is not known with certainty. Furthermore, the travel times of customers on the same route vary. Under these circumstances of randomness coupled with parameter uncertainty, it is desirable to synchronize the traffic lights in such a way that traffic does not accumulate at a particular road segment and the traffic empties from the grid as soon as possible, see Figure (1.1). The job shop framework we develop models this situation. In case of the control of traffic lights, a natural choice is a non-adaptive policy where, at each traffic light the cycle lengths and percentage allocations for each direction to have a green light are decided a priori and not modified.

Building on the above motivating example, we model a job shop where

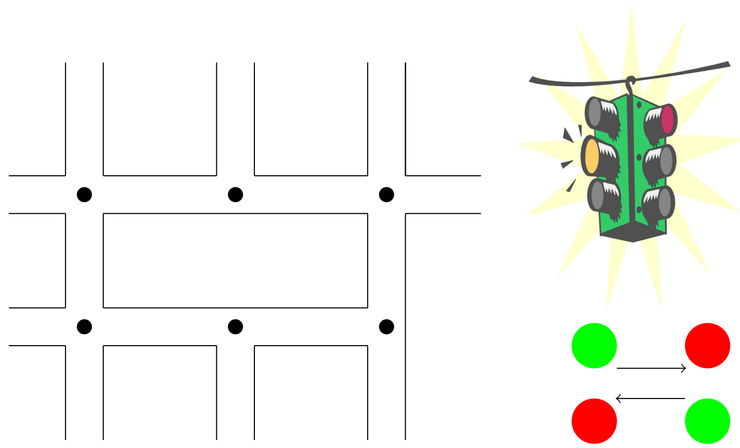


Figure 1.1: Signal control- A grid

processing times of jobs are random and the number of jobs is uncertain. As in the control of traffic lights, we considered a class of policies (cyclic policies) in which at each server, the manager decides cycle lengths and within each cycle, the percentage of time allocated to processing each job class in a cycle. The scheduling decisions must be made a priori. The objective is to minimize expected makespan, i.e., the total expected time taken to complete all the jobs. Minimizing the expected makespan can be seen as a first step towards developing a generic modeling framework. The makespan problem is easier to analyze than, for example, the holding cost problem, and promises to offer valuable insight into the solution structure in general. For cyclic policies, when certain integrality requirements are dropped and processing times are replaced by their means, we show that the job shop scheduling problem reduces to a 'fluid model.' The optimization problem for the fluid model is a stochastic non-linear program without integer constraints and is easier to solve than the original discrete job shop problem. Based on the optimal solution to the fluid model a scheduling heuristic for the job shop is proposed. We prove the asymptotic optimality of the heuristic, as the number of jobs grows large. We analyze different instances of this job shop model and modify the heuristic accordingly.

Next we provide descriptions of the structure of control decisions and models, assumptions and examples. In the section below we also provide a high level literature review. In later chapters we introduce more detailed background as needed.

1.1 Service Rate and Admission Control

The $M/M/1$ system with both admission and service rate controls is an idealized model for practical scenarios; nonetheless it may offer valuable insight. We can conceive a number of practical scenarios where the model is appropriate. One such instance is in context of a static wireless link as suggested by Ata [1], where the controller chooses state-dependent transmission rate subject to a quality-of-service (QoS) constraint. However, in that model the buffer size is fixed. This power control problem can be extended, under less imposing assumptions, to include cost associated with QoS, i.e., a drop rate. This allows us to the model buffer size as a control decision. A control strategy involving admission and service rate controls, is also applicable in service sector applications such as a call center where dynamic staffing is possible and additionally calls can be transferred to an alternate more costly facility at a fixed cost per call. As discussed previously, the same setup can be used to model a single manufacturing facility, with different in-house and outsourcing options.

We briefly introduce our model here in order to facilitate discussion of related literature. The model consists of a single server with adjustable service rate. The service time of a customer being served at a constant rate $x > 0$,

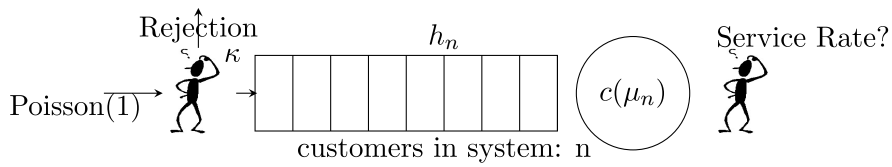


Figure 1.2: Dual Control

is exponentially distributed with mean $1/x$. It is assumed that arrivals occur according to a Poisson process. Without loss of generality we take the rate of this process to be 1. The system manager can change the service rate at any instance of time. Further, the arrivals can be denied admission by the system manager. There are costs associated both with providing service and denying admission. These costs along with the holding costs comprise the total system cost. The objective of the system manager is to minimize the long run average cost per unit time.

All costs or cost functions are known to the system manager. Let the cost for service be $c(x)$ for a service rate x, and let h_n be the holding cost for a queue length n. An important aspect of the model is inclusion of rejection combined with service rate control. Rejecting a customer request is the same as providing instantaneous service. The cost incurred for rejecting a customer is κ. A holding cost rate of h_0 (when no queue is present), is incurred even if the customer is rejected service. The system dynamics are represented in Figure 1.2. The figure depicts a policy with state-dependant deterministic control. The system manager has to choose a system threshold beyond which, customers are rejected service and, state-dependent service rates.

The $M/M/1$ control problem with service rate or admission decisions has been addressed by many authors in various forms. However, the combined problem with both controls has not been sufficiently studied in literature. As stated in [27], the $M/M/1$ queue is probably the most studied queueing model. Hence, it is quite natural that we look at the combined control problem for this simple system. We allow the set of available service rates to be unbounded which is different from the action space assumption in many standard models found in the Markov decision process (MDP) literature such as [3, 23] and [27]. In [15], Guo and Hernández consider the overall action space to be unbounded, however, for each state, the action space is bounded. In [13] unbounded service rates are modeled within the MDP framework. Their model was further modified by Ata and Shneorson [2] to include capacity constraints. [13] provides a comprehensive list of related work, both MDP analysis and computation. Usually computation methodologies for MDPs like policy iteration or value iteration are too generic and cannot be applied to the problem we consider due to the requirement of a bounded action space. Exploiting the structure of our model, we develop a computation methodology. It is worthwhile to note that none of the previous papers establish the existence of a stationary optimal policy when the action space is unbounded.

The set of non-anticipatory controls, i.e., history dependent randomized policies (HR) are considered, see [23] for background. However, initially we take into account only stationary controls. These controls are deterministic and state-dependent. A policy $(\vec{\mu}, m)$ consists of two elements: a vector of state-dependent service rates μ_n and a threshold on the maximum number customers in the system m beyond which customers are rejected service. We assume the policies to be ergodic. For a given policy, at any particular queue length, a customer can either be served at a fixed finite service rate or rejected service. We construct optimality equations from first principles. The existence of a optimal stationary policy is proved by demonstrating the validity of these equations in the context of Markovian randomized policies (MR).

The computation method we suggest takes advantage of the model structure and simplicity. The iterative method is two-fold, with the first stage decision being the maximum allowed buffer size beyond which all arriving customers are rejected. Thereafter through a convergent iterative algorithm, the optimal service rates for each state and consequently the optimal average cost, for the fixed buffer size are determined. The buffer size is increased in increments of unit size till a local minimum in terms of average optimal cost is achieved. We prove that the policy associated with the first local cost minimum is optimal among the class of stationary policies. In the absence of local minima, we establish that a limiting ergodic policy exists where customers are not rejected and the limiting policy is optimal among the class of stationary policies. Further, we prove the optimal policy is monotone with respect to the system state.

1.2 Stochastic Scheduling under Uncertainty

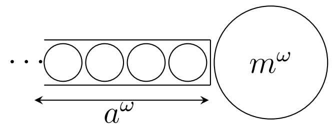
In many systems, in addition to the inherent stochastic nature of processing times or arrival times, parameters like server capacity are not known with certainty. A typical instance is that of a manufacturing plant subject to a sudden disruption. The amount of unfinished product inventory, as well as the operating conditions of servers might be uncertain. Further, the processing times for an operation might be random, i.e., they may vary from one job to the next. In the spirit of the motivating example above, we model a job shop with an uncertain number of initial jobs and doubly stochastic processing times. To make the analysis tractable, we restrict the scheduling to a specific class of policies: cyclic policies. However, the choice of the class of policies is not without rationale. One example might be industries where workforce restrictions are stringent and fixed schedules are the norm. For the makespan objective, we obtain a relaxed version of the scheduling problem. We suggest asymptotically optimal heuristics based on the optimal solution of a relaxed version. The main contributions of this work pertaining to job shop scheduling are:

1. building a framework to incorporate parameter uncertainty,

2. developing asymptotically optimal heuristics.

Job shop scheduling has been widely studied in the literature. Even scheduling with the seemingly simple objective of minimizing makespan in a completely deterministic job shop is NP-hard. When there are large number of identical jobs of different types, the job shop is well-modeled by multi-type queueing network (MTQN). Recently there has been much interest in developing scheduling heuristics for such job shops based on the optimal solution of an associated fluid model. In this analysis, we consider this kind of job shop with high multiplicity of jobs and its associated fluid model. The job shop described above consists of a fixed set of job types. A job type is associated with a fixed sequence of steps through the job shop, i.e., there is a fixed route. Each job belongs to one of the job types. A combination of job type and a particular step in the processing sequence is known as job class.

The framework we introduce is 'two-dimensional': **1**. scenario dependent parameters and, **2**. time evolving randomness in each scenario. Figure (1.3) is a representation of this framework. In the single server example depicted in the figure, uncertainty is associated with the number of jobs as well



ω	a	m
1	10	$U(1,3)$
2	5	$U(1,5)$

Figure 1.3: Single Server: 'Two-dimensional framework'

as the mean processing times. Two scenarios, ω_1 and ω_2, are possible. Depending on which scenario unfolds, the parameters (a^ω, m^ω) are realized. In each scenario the processing time for each job is a uniform random variable. For example, in the second scenario, each of the 5 jobs has a random processing time uniformly distributed on the interval $(1, 5)$. We assume that the parameter randomness is realized at the initial time say $t = t_0$. Further, we assume the scheduler has an idea (the probability distribution) regarding the parameter uncertainty. This kind of job shop is called as stochastic job shop with parameter uncertainty. Note that the associated objective functions and dynamics are to be analyzed in a probabilistic setting. In this work, abusing the nomenclature, we use the term *stochastic job shop* for the job shop with parameter uncertainty.

The authors Bertsimas, Gamarnik and Sethuraman, in a series of papers [4, 5] and [6] consider a multi-type job shop as described above. In their models all quantities and parameters are assumed to be deterministic. Both the makespan and weighted holding cost objective are considered. These papers use fluid models of the job shop to develop "fluid following" heuristics and show that the heuristics are asymptotically optimal. In [12], Dai considers makespan minimization with random processing times. They develop a heuristic scheduling method which is shown to perform well with high probability. A similar adaptive policy is developed by Maglaras in [21]. This policy compares the fluid model schedule with that of job shop after discrete time-slots and makes modifications. However, unlike our fluid model, the fluid model the author considers is a limiting model to a sequence of job shops under fluid scaling. This policy is modified to facilitate continuous tracking (adaption) in [22].

Cyclic policies have been studied in the literature especially in relation to scheduling of FMS (flexible manufacturing system). For example in [18], the authors suggest a search algorithm to develop a good cyclic policy. However, parameter uncertainty is not present in the model. Cyclic schedules for job shop with a single job type are studied by Roundy [25]. Job release in conjunction with cyclic schedules for the smooth operation of the job shop is discussed by Bowman in [7]. As in the present work, in [4] the authors propose cyclic policies with the possibility for idling and show that the policy is asymptotically optimal for makespan minimization for a deterministic job shop.

There is abundant literature addressing the question of developing optimal or near-optimal controls for the fluid model in the more generic setting of multi-class queueing networks (MCQN). One closely related work is by Weiss [29], in which, the author derives an analytical closed-form solution for the optimal time allocation in the fluid draining problem, when the parameters are known. The makespan problem for MCQN with parameter uncertainty is introduced and studied by Burak et al. [8]. The makespan problem for our fluid model falls under the framework developed in [8].

We suggest heuristics for a stochastic job shop with parameter uncertainty, based on an optimal solution for the makespan problem of the associated fluid model. Equivalently, for cyclic policies, the heuristics specifies the allocation of stations' capacity among job classes. Optimal draining for the fluid model is the analog of makespan minimization for discrete job shop scheduling problem. The optimization problem for the fluid model is a stochastic non-linear program, without integer constraints, which is easier to solve. We also consider the specific case of a single station, where the objective to assign servers to job classes. In the fluid following policy, we assign servers to match the allocation in an optimal fluid solution as closely as possible.

We show that the heuristics constructed in the manner described above are asymptotically optimal, i.e., they are near optimal when the initial number of jobs in the system is large. When the processing times are random, we demonstrate asymptotic optimality in terms of the expected makespan. We also show that for each scenario, with high probability, the makespan lies within tight bounds of that of the associated fluid solution.

The organization of various chapters related to the first model is as follows. In Chapter 2 the problem is formulated in its mathematical form. We provide optimality equations and verify them. Also, using the theory of semi-Markov processes the existence of a stationary optimal policy is proved. Next the computation method is developed in Chapter 3. Finally a couple of numerical examples are provided in Chapter 4.

In Chapter 5 we introduce the various entities and notation related to the problem. We formulate a generic job shop scheduling problem under makespan minimization. We also, mention the various policy sets studied in later chapters and under specific policy settings modify the model. We derive the associated fluid models for these policy sets. Next, in Chapter 6 the single station job shop without reentry is analyzed extensively. Two sets of policy settings, assignment policies and cyclic policies, are studied. We propose fluid following policies heuristics and prove them to be asymptotically optimal under the right setting, depending on the model. In Chapter 7 we extend this analysis to a generic job shop. Both cases, when processing times are doubly stochastic or just scenario dependent, are analyzed in Chapters 6 and 7.

Chapter 2

Service Rate and Admission Control

The control problem associated with the single server system is to find an optimal control to minimize the average long run cost per unit time. In this chapter we develop a set of optimality equations for stationary policies and show that a solution to this set of equations provides a lower bound on the optimal average cost. We construct a policy from this solution and show that if this policy is ergodic and the problem is non-degenerate then the constructed policy is an optimal policy. Further, we also show that these results hold when average cost optimization is considered over all non-anticipatory policies.

The decision framework for the problem is a continuous time Markov decision process with countable state space. Under a stationary control the associated process is continuous time Markov process. Until Section 2.2 the analysis is restricted to stationary controls.

We assume the following:

A1 The action space A is a closed subset on $[0, \infty)$ containing 0 and an element greater than 1.

A2 The holding cost h_n is non-decreasing in n.

A3 The cost for service function $c(x)$ is non-decreasing in x.

A4 $c(x)$ is left continuous with $c(0) = 0$.

These assumptions are appealing in a practical sense and are standard ones, simplifying the analysis. When A is unbounded the assumption in which this discussion differs from [13] and allows for the inclusion of rejection is

A5 $\lim_{y\to\infty} \left(\inf \left\{ \frac{c(x)}{x} : x \in A, x \geq y \right\}\right) \geq \kappa \geq 0.$

Assumption **A5** implies that the cost for service per unit time is greater than the rejection cost, as the service rate grows without bound. When **A5** is not satisfied, the option of admission control can be ignored, as serving at some large service rate is always beneficial as compared to rejecting the customer. The above assumptions are the only ones considered in the analysis.

The admission control problem for a system with fixed arrival and service rates is treated in [23]. The state transitions diagram for this problem is similar to that of the model under consideration, under an ergodic MR policy. The only difference is presence of variable service rates. The admission control decision is specified by a, where $a \in \{0, 1\}$. $a = 1$ when the decision is to admit the new customer. A service rate can be changed when the system state changes due to a departure or an arrival, even if the customer is not admitted. On an arrival the control decision can be represented as (a, x), where the service rate until the next transition is $0 \leq x < \infty$. When the state changes due to a departure, the control decision is equivalent to that when a system states changes due to arrival and $a = 1$. Without loss of generality we assume $x = 0$ when $n = 0$.

For stationary ergodic policy a single service rate and admission decision is associated with each state, i.e., (a, x) is state dependent. To reflect state dependent nature of the controls, representation of a stationary policy is changed as described below. Under a stationary ergodic policy $(\vec{\mu}, m)$, all customers arriving when the system state is m, the system threshold, are rejected service. When there are n customers in the system the server processes jobs at rate μ_n. For an ergodic policy-$(\vec{\mu}, m)$ the long run cost per unit time is given by

$$z(\vec{\mu}, m) = p_m(\vec{\mu}, m)\kappa + \sum_{n=0}^{m} p_n(\vec{\mu}, m)\left\{c(\mu_n) + h_n\right\} \tag{2.1}$$

where $p_n(\vec{\mu}, m)$ is the steady state probability that the queue length is n and μ_n is the service rate under the policy $(\vec{\mu}, m)$ when the queue length is $n \leq m$.

We allow m to take the value infinity. When m is infinite, the first term on the right hand side of (2.1) is absent and the sum in the second term is an infinite sum. For convenience, a finite m is defined to be *terminating state* and the associated policy is *terminating*. Further, the policy with terminating state as m, is said to be m terminating. The relation between the probabilities is given by the usual local balance equations in a birth-death process:

$$p_n(\vec{\mu}, m) = p_{n-1}(\vec{\mu}, m)\mu_n^{-1} \quad \forall \quad 1 \leq n \leq m, \mu_n > 0. \tag{2.2}$$

When no service is offered, i.e., $\mu_n = 0$ for some n, the modification of the probabilities is straightforward. For ergodicity of a service policy with no rejection, i.e., when m is infinite, the number of service states with $\mu_n = 0$ has to be finite. If customers are rejected in every state, the resulting policy is ergodic with $z(0,0) = h_0 + \kappa < \infty$. Thus, there exists at least one ergodic policy with a finite average cost. A more generic ergodic policy can be easily constructed. As a result, the infimum of the long-run average cost is well defined and finite. Define

$$z^*(m) = \inf z(\vec{\mu}, m),$$

where the infimum is taken over all m terminating policies. Note that $z(0,0) = z^*(0)$. A policy with $z(\vec{\mu}, m) = z^*(m)$, if it exists, is called m-optimal. The infimum across all ergodic policies is:

$$z^* = \inf \{m \geq 0 : z^*(m)\}.$$

The control problem is to find a ergodic policy $(\vec{\mu}, m)$ which achieves the infimum. A policy is said to be optimal if it achieves this infimum. In general for a MDP, existence of an optimal policy among the class of stationary policies is not guaranteed. However, for the present control problem existence is proved through the computation method, i.e., the algorithm either converges to a stationary ergodic policy. This is shown in Chapter 3. When holding cost rates are bounded, it is possible that $h_n \uparrow h_\infty \leq z^* < \infty$, i.e., the long run average cost under the *do-nothing policy* is smaller than achievable long-run average cost under any ergodic policy. As in [13], in this case the dynamic program is said to be *degenerate*.

The rest of this chapter is organized as follows. In Section 2.1 we provide a crude derivation of the optimality equations and, rigourously verify the validity of the reduced form of the optimality equations. Also, for a system with fixed threshold modified optimality equations are provided. In Section 2.2 we analyze the control problem in the context of Markovian randomized policies. We show that the original optimality equations also hold when the associated semi-Markov processes (SMP) are considered. Finally it is argued that an optimal stationary policy exists and that, the computation algorithm given in Chapter 3 terminates or converges to an optimal policy.

2.1 Optimality Equation

Using standard MDP arguments we write the basic optimality equations:

$$v_n = \min \begin{pmatrix} \inf\limits_{x\in A} \left\{ \frac{c(x)+h_n-z+xv_{n-1}+v_{n+1}}{1+x} \right\}, \\ \inf\limits_{x\in A} \left\{ \frac{c(x)+h_n-z+xv_{n-1}+v_n+\kappa}{1+x} \right\} \end{pmatrix} \qquad \forall \quad n \geq 1,$$

$$\Rightarrow 0 = \min \begin{pmatrix} \inf\limits_{x\in A} \left\{ \frac{c(x)+h_n-z-x(v_n-v_{n-1})+(v_{n+1}-v_n)}{1+x} \right\}, \\ \inf\limits_{x\in A} \left\{ \frac{c(x)+h_n-z-x(v_n-v_{n-1})+\kappa}{1+x} \right\} \end{pmatrix} \qquad \forall \quad n \geq 1 \tag{2.3a}$$

and

$$v_1 = v_0 - h_0 + z. \tag{2.3b}$$

Following the arguments presented in [13] and [30] and substituting the *relative cost difference* functions with *relative cost* functions, (2.3a)–(2.3b) reduce to the following form

$$z - h_n = \min \begin{pmatrix} \inf\limits_{x \in A} \{c(x) - y_n x + y_{n+1}\} \\ \inf\limits_{x \in A} \{c(x) - y_n x + \kappa\} \end{pmatrix} \quad \forall \quad n \geq 1, \tag{2.4a}$$

and

$$y_1 = z - h_0, \tag{2.4b}$$

where $y_n = v_n - v_{n-1}$ for $n \geq 1$. Defining,

$$\phi(y) = \sup_{x \in A} \{yx - c(x)\}, \qquad \text{for } y \geq 0, \tag{2.5}$$

simplifies the optimality equations, specifically (2.4a) becomes

$$h_n - z = \phi(y_n) - \min\{y_{n+1}, \kappa\} \quad \forall \quad n \geq 1. \tag{2.6}$$

The first value of n for which the second term in (2.6) is a minimum, i.e., $y_{n+1} \geq \kappa$, is said to be the terminating state for the optimality equations. This corresponds to the terminating state m of the associated policy. If $y_{n+1} \leq \kappa$ for all $n \geq 1$, then the solution pair is said to be non-terminating.

Note that if a particular pair z and $(y_1, y_2, \dots)$ is a solution of the optimality equations (2.4b) and (2.6), then $y_{n+1} < \kappa$ for any non-terminating state $n \geq 1$. When $y < \kappa$, under **A1**, **A4** and **A5**, the function $\phi(y)$ has finite values which are attained and the smallest maximizers in the set A exist. $\psi(y)$ is the smallest minimizer of $\phi(y)$. Note that assumption **A5** implies that the smallest $\psi(y)$ is finite. Further, if $\phi(\kappa) < \infty$, the smallest minimizer $\psi(\kappa)$ is well defined.

It is worthwhile to note that the optimality equations remain the same even if the holding cost is not non-decreasing. In the next subsection the formal proof for the optimality equations is given. Also, a modified version of these optimality equations is introduced and verified.

2.1.1 Verification Theorem

Theorem 1. *Let $z < \infty$, $(y_1, y_2, \dots)$ with each element uniformly bounded be a solution to the optimality equations (2.4b) and (2.6), and m^* the corresponding terminating state. Then $z \leq z(\vec{\mu}, m)$ for every ergodic policy $(\vec{\mu}, m)$, i.e. $z \leq z^*$. If the policy $(\vec{\mu}^*, m^*)$ defined by*

$$\mu_n^* = \begin{cases} \psi(y_n) & \text{if } 1 \leq n < m^*; \\ \psi(y_{m^*}) & \text{if } n \geq m^* < \infty \end{cases}$$

is ergodic, then (μ^*, m^*) is optimal.

Proof. First, note that since $z < \infty$ and the y_i are bounded, $\phi(y_n) < \infty$ for all $n \geq 1$. Any ergodic policy is either a terminating or non-terminating policy. In either case the following relations hold, from definition (2.5)

$$xy_n - c(x) \leq \phi(y_n) \leq y_{n+1} + h_n - z \quad \forall \quad x \geq 0, n \geq 1, \tag{2.7}$$

$$xy_n - c(x) \leq \phi(y_n) \leq \kappa - h_n - z \quad \forall \quad x \geq 0, n \geq 1. \tag{2.8}$$

For any arbitrary terminating policy $(\vec{\mu}, m)$, setting $x = \mu_n$ in (2.7) and $x = \mu_m$ for $n = m$ in (2.8), we have

$$\mu_n y_n - c(\mu_n) \leq y_{n+1} + h_n - z \qquad \text{for } n \geq 1, \tag{2.9}$$

$$\mu_m y_m - c(\mu_m) - \kappa \leq h_m - z \qquad \text{for } m < \infty. \tag{2.10}$$

Multiplying both sides of (2.9) by $p_n(\vec{\mu}, m)$, multiplying both sides of (2.10) by $p_m(\vec{\mu}, m)$, substituting $\mu_n p_n(\vec{\mu}, m) = p_{n-1}(\vec{\mu}, m)$ from (2.2), and rearranging terms for $m > n \geq 1$, we have

$$p_n(\vec{\mu}, m)[h_n + c(\mu_n) - z] \geq p_{n-1}(\vec{\mu}, m)y_n - p_n(\vec{\mu}, m)y_{n+1} \tag{2.11a}$$

$$p_m(\vec{\mu}, m)\,[h_m + c(\mu_m) + \kappa - z] \geq p_{m-1}(\vec{\mu}, m)y_m. \tag{2.11b}$$

Summing over all n for $m > n \geq 1$ and, using relations (2.1) and (2.4b),

$$\begin{aligned} z(\vec{\mu}, m) - p_0(\vec{\mu}, m)h_0 - z[1 - p_0(\vec{\mu}, m)] &\geq p_0(\vec{\mu}, m)y_1 \\ \Rightarrow z(\vec{\mu}, m) &\geq z. \end{aligned} \tag{2.12}$$

For a non-terminating policy $(\vec{\mu}, \infty)$, the summation (2.11a) is over all $n \geq 1$. Since the y_i are bounded the sum is finite and the above result is valid. Hence the result $z(\vec{\mu}, \infty) \geq z$ holds.

When we set $x = \psi(y_n)$ in (2.9) and $x = \psi(y^*_m)$ in (2.10) are set, then (2.9) and (2.10) hold with equality. Hence equations (2.11a)-(2.11b) hold with equality when the ergodic policy with $\mu_n = \mu^*_n = \psi(y_n)$ for $1 \leq n < m^*$ and $\mu_{m^*} = \mu^*_{m^*} = \psi(y_{m^*})$, if $m^* < \infty$, is employed. As a result (2.12) holds with equality for this policy, i.e., (μ^*, m^*) is optimal. □

In the above result, for completeness, even for states beyond the terminating state, a service rate is assigned. These states are transient states.

Below the modified optimality equations are given when only n-terminating policy are considered for a fixed n. These are also called n-optimality equations. The following theorem is stated without a proof, as it follows directly from the above proof.

Theorem 2. *If there exists an $n \geq 0$ a sequence $(y_1, \cdots, y_n)$ and $z < \infty$, satisfying (2.4b) and*

$$\begin{aligned} h_k - z &= \phi(y_k) - y_{k+1} \quad \textit{for } 1 \leq k \leq n \\ y_{n+1} &= \kappa \end{aligned} \tag{2.13}$$

then $z^*(n) = z$ and the policy $(\vec{\mu}^*, n)$ given by

$$\mu_k^* = \psi(y_k) \quad \textit{for } 1 \leq k \leq n \tag{2.14}$$

is n-optimal.

This theorem can be applied when a system with a fixed threshold is considered and there is no option of rejecting customers unless the buffer is full. One such case is discussed in [2]. Note that both proofs hold even when the holding cost h_n is not non-decreasing is n.

2.2 Stationarity

In general, there need not exist a stationary optimal policy in a given MDP. In the present control problem, if the existence of a stationary optimal policy is guaranteed then it suffices to restrict the analysis to such policies. Also, the presence of potentially unbounded service rates and unbounded costs makes the question of existence of a stationary optimal policy more interesting. We are not aware of any previous work establishing the existence of a stationary optimal policy when the action space is unbounded. Control problems with adjustable service and service rates along with additional constraints are discussed in [15, Section 6] and existence of an optimal stationary policy is established. However, there it is assumed that the service rates are bounded.

When any arbitrary policy is considered, at any point of time the control can be considered to have two components: admission control and service rate control. Service rate control can be seen to be feasible only when a customer is present in the system and admission control feasible only when a new customer arrives. Note that it suffices to consider just MR policies, see Theorem 11.1.1 in [23]. However, when a MR policy is considered, the control rate is random in any particular state. So, the associated process is a continuous time SMP. Using SMP techniques, we prove the validity of the original optimality equations in the context of MR policies. We show that if the policy $(\vec{\mu}^*, m^*)$ defined in Theorem 1 is ergodic then it is optimal, provided the problem is not degenerate. We also argue that the analysis in Chapter 3 holds and so even if a solution to the optimality equations does not exist, the stationary optimal policy is the one specified in Theorems 4 and 6.

Further, Markovian randomization of the decision in each state is allowed. For each system state n, the randomization consists of three components $(q_n, F_n(1, x), F_{n-1}(0, x))$. q_n is the probability of accepting a customer, and the choice of service rate x, is made according to the cdf $F_{n-1+a}(a, x)$ depending on the admission decision a. After a departure the service rate choice is made according to the distribution $F_n(1, x)$. The average cost of rejecting a customer κ, is independent of the service rate chosen after rejecting the customer. Hence it is vital to consider the service rate distribution functions. Equivalently when the number of customers in the system is $n \geq 0$, before an arrival or after a departure, the randomized policy can be specified by the vector $\vec{Q} \equiv \{Q_n\}$, where $Q_n \equiv (q_n, F_n(0, \cdot), F_n(1, \cdot))$

In the resulting system with MR policies, the service time in a state n follows a general distribution which is determined by Q_n. It should be clear that the queue length process is then a semi-Markov process. From results on semi-Markov process (SMP) given in [14], the following relations hold. ν_n, the steady state probability that the SMP is in state n satisfies

$$\nu_n = \frac{\pi_n \vartheta_n}{\sum\limits_{i=0}^{\infty} \pi_i \vartheta_i} \tag{2.15}$$

where ϑ_n is the mean time spent in the state n during each visit, until the next transition and π_n is the stationary probability of being in state n in the embedded Markov chain. For (2.15) to hold the transitions times ϑ_n for all states $n \in \mathbb{N}$, must be finite and have non-lattice distribution. Since we are considering ergodic policies, without loss of generality we can assume that the mean time to transition between communicating states n and $n+1$ is finite. . So in case of out problem the required conditions for (2.15) to be valid are satisfied. Define $G_n(t)$ to be the cdf of the time to the next transition when in the state n. $\mathcal{E}_n[\cdot]$ denotes expectation with respect to the randomization tuple Q_n, i.e., expectation when the distribution function is

$q_n F_n(1,x) + (1-q_n)F_n(0,x)$. Then

$$\begin{aligned} G_n(t) &\equiv \int_0^\infty \left[1 - e^{-(1+x)t}\right] d\left(q_n F_n(1,x) + (1-q_n)F_n(0,x)\right) \\ &= \mathcal{E}_n\left[1 - e^{-(1+\mathcal{X})t}\right], \end{aligned}$$

where $\mathcal{X}$ is the random service rate. $m_n = \int_0^\infty t dG_n(t)$, which implies

$$\begin{aligned} m_n &= \mathcal{E}_n\left[\int_0^\infty (1+\mathcal{X})te^{-(1+\mathcal{X})t} dt\right] \\ &= \mathcal{E}_n\left[\frac{1}{1+\mathcal{X}}\right] \\ &= \frac{1}{1+\bar{\mu}_n}, \end{aligned} \tag{2.16}$$

where $\bar{\mu}_n \leq \infty$ is defined by the last equality. From (2.16), using Jensen's inequality we have

$$\begin{aligned} \mathcal{E}_n\left[\frac{1}{1+\mathcal{X}}\right] &\geq \frac{1}{1+\mathcal{E}_n[\mathcal{X}]} \\ &\Rightarrow \mathcal{E}_n[\mathcal{X}] \geq \bar{\mu}_n. \end{aligned} \tag{2.17}$$

Now with the above inequalities, Theorem 1 can be extended to include MR policies. This is formalized below. The cost of any MR policy $\vec{Q}$, is

$$z(\vec{Q}) = \nu_0 h_0 + \sum_{n=1}^{\infty} \nu_n \left[h_n + \mathcal{E}_n[c(\mathcal{X})] + (1-q_n)\kappa\right]. \tag{2.18}$$

Theorem 3. *If there exist $z < \infty$ and uniformly bounded $(y_1, y_2, \dots)$, satisfy-*

ing the optimality equations (2.4b)-(2.6), and the resulting policy as specified in Theorem 1 is ergodic, then $z = z^* \leq z(Q)$ for every MR policy $\vec{Q}$.

Proof. Define $\tilde{\mu}_n := \mathcal{E}_n[\mathcal{X}]$. From equations (2.7)–(2.8), taking expectation we get

$$xy_n - c(x) \leq \ \phi(y_n) \leq y_{n+1} + h_n - z \quad \forall \quad x \geq 0, n \geq 1,$$

$$xy_n - c(x) \leq \ \phi(y_n) \leq \kappa + h_n - z \qquad \forall \quad x \geq 0, n \geq 1.$$

$$\Rightarrow y_n\tilde{\mu}_n - \mathcal{E}_n[c(\mathcal{X})] - (1-q_n)\kappa - y_{n+1}q_n \leq h_n - z^*. \tag{2.19}$$

For the transition probabilities of the embedded Markov chain we have

$$p_{n,l} = \begin{cases} \frac{1}{1+\tilde{\mu}_n} q_n & \text{if } n \geq 1, l = n+1; \\ \frac{1}{1+\tilde{\mu}_n}(1-q_n) & \text{if } n \geq 1, l = n; \\ \frac{\tilde{\mu}_n}{1+\tilde{\mu}_n} & \text{if } n \geq 1, l = n-1; \\ q_0 & \text{if } n = 0, l = 1; \\ 1 - q_0 & \text{if } n = l = 0; \\ 0 & \text{o.w.} \end{cases}$$

and

$$\pi_n p_{n,n+1} = \pi_{n+1} p_{n+1,n} \quad \text{for } n \geq 0$$

which imply

$$q_n \nu_n = \nu_{n+1}\tilde{\mu}_{n+1} \quad \text{for } n \geq 0 \tag{2.20}$$

Combining the simple relation (2.20) between the steady state probabilities, and (2.19) for $n \geq 1$, we have

$$\begin{aligned} & y_n \nu_n \tilde{\mu}_n - \nu_n \mathcal{E}_n[c(\mathcal{X})] - \nu_n(1-q_n)\kappa - \nu_n q_n y_{n+1} \leq h_n - z^* \\ \Rightarrow\ & y_n q_{n-1} \nu_{n-1} - y_{n+1} q_n \nu_n - \nu_n \mathcal{E}_n[c(\mathcal{X})] - \nu_n(1-q_n)\kappa \leq \nu_n[h_n - z^*]. \quad (2.21) \end{aligned}$$

Since the y_i are finite, summing over all $n \geq 1$ as in Theorem 1 the result follows. Also, arguing as in Theorem 1, if $(\vec{\mu}^*, m^*)$ is ergodic, then it is optimal. □

From first principles we have proved the existence of a stationary optimal policy, if a solution of the optimality equations exists and the problem is non-degenerate. The result is derived from the balance equations for the semi-Markov process and the original optimality equation. However, Theorem 3 in itself does not guarantee the existence of a stationary optimal policy, as a solution pair of the optimality equations need not exist. If the sequence of terminating optimal policies has a locally optimal, then, as argued in Theorem 4 this local optimum policy satisfies the optimality equations for a system with a modified holding cost structure, where beyond certain state the cost rates are set uniformly to a value lower than all the original values. The modified holding costs are lower than the original holding costs. For this system, there exists an optimal stationary policy from Theorem 3. Hence it follows that this local optimum policy is optimal for the original system among the class of HR policies. In case the sequence of terminating policies is decreasing, the limiting policy satisfies the optimality equations of the original system. This implies that the limiting policy is optimal among the class of HR policies.

The argument above proves that even if a solution pair for the optimality equations does not exist, an optimal stationary policy exists. This methodology can probably be modified to address an array of control problems, for example when the service rate can be varied as in [2].

Chapter 3

Computation of the Optimal Policy

We have developed optimality equations and established their validity in the previous chapter. In this chapter, we suggest a computation methodology to achieve an optimal or near optimal policy. The optimality equations alone do not suggest an efficient methodology for constructing near optimal policies. The methodology suggested in this chapter helps find a optimal policy even when a solution to the optimality equations does not exist. The methodology is two-fold, as follows

Initialize Set $n = 1$.

Step 1 Solve the $n-$terminating optimality equations.

Step 2 If a solution exists and $z^*(n) \geq z^*(n-1)$ then, the $(n-1)$-optimal policy is an optimal policy for the original system. Otherwise, the threshold n is incremented to $n+1$ and steps 1 and 2 are repeated.

Consider the sequence of solutions $(z(n), (y_1^n, \cdots, y_n^n))$ satisfying equations (2.4b) and (2.13) for $n \geq 1$. If a solution pair exists for $n \geq 1$ then Theorem 1 applies, i.e., $z(n) = z^*(n)$. The existence of a solution pair is not guaranteed. The n^{th} solution pair corresponds to an optimal policy for a system with a fixed threshold n. The sequence of resulting policies is that of terminating optimal policies, increasing in terms of the terminating state. Starting from $n = 1$ solution pairs are to be computed for incremental values of n. As mentioned above, this computation continues till a local minimum in terms of $z^*(n)$ is found or no solution pair exists. It is shown that the first local minimum of the sequence of $z(n)$ is equal to z^*, otherwise $z(n)$ converges to z^*. The sequence of terminating optimal policies can be seen as corresponding to progressive approximations of a optimal policy.

In Section 3.1 we prove optimality of the first local minimum of the sequence of terminating policies described above. Note that since this method is based on terminating policies and stops at a local minimum, we are not concerned with the existence of a solution to the optimality equations. Section 3.2 discusses convergence of the sequence of terminating policies when they are decreasing in terms of average cost. The limiting policy is shown to be an optimal policy. We show that under both scenarios the constructed optimal policies are monotone in terms of the number of customers in the system.

3.1 Optimal Policy: Sequence with a minimum

The sequence of solutions sequence of solutions $(z(1), y_1^1), (z(2), y_1^2, y_2^2), \ldots$ might have a local minimum or terminate when no solution pair exist for a particular $n+1$. When no solution exists it is shown that $z(n) < z^*(n+1)$. Hence, in both cases the sequence is said to have a local minimum. Here, we show that when $z(n)$ is the first local minimum of the sequence, then the n-optimal policy is an optimal policy. This policy is shown to be monotone in terms of the state. Monotonicity in terms of the service rates for progressive approximations is also established.

Lemma 1. *If $z(k) > z(m)$ for $m > k \geq 1$, then*

$$y_{k+1}^m < \kappa. \tag{3.1}$$

Proof. Since $z(k) > z(m)$, we have

$$\begin{aligned} h_k - z(m) &> h_k - z(k) \\ \Rightarrow \phi(y_k^m) - y_{k+1}^m &> \phi(y_k^k) - \kappa, \end{aligned} \tag{3.2}$$

where the later inequality is due to (2.13). From the construction of the sequence, $y_k^{(n)}$ is a non-decreasing function in $z(n), n \geq k$. This implies that $y_k^k > y_k^m$, as $z(k) > z(m)$. Since $\phi(y)$ is also an non-decreasing function, from (3.2

$$\phi(y_k^k) \geq \phi(y_k^m)$$
$$\Rightarrow y_{k+1}^m < \kappa. \qquad \square$$

For a decreasing sequence of $z's$, up till a local minimum, the Lemma 1 establishes that $y's$ are bounded away from κ, i.e., they satisfy the property of non-terminating states. The following theorem is the main result of this subsection.

Theorem 4. *If there exists a sequence of terminating optimal solutions, satisfying*

$$\begin{aligned} z(k) &> z(m) \quad \textit{for } m > k \geq 0, \\ z^*(m+1) &\geq z(m) \end{aligned} \tag{3.3}$$

then the m-optimal policy given by the solution $(z(m), (y_1^m, \cdots, y_m^m))$ is optimal. Also

$$y_1^m \geq y_2^m \geq \cdots \geq y_m^m. \tag{3.4}$$

Proof. Note that $m \geq 1$. The non-decreasing nature of y_k^n, in terms of k, for $1 \leq n \leq m$ is due to the cost structure. Since the holding cost rate is non-decreasing, from equation (2.13),

$$
\begin{aligned}
\phi(y_n^n) - \kappa &= h_n - z(n) \\
&\geq h_{n-1} - z(n) \quad \text{ for } 1 < n \leq m \\
&\geq \phi(y_{n-1}^n) - \kappa \\
\Rightarrow y_n^n &\leq y_{n-1}^n \\
\Rightarrow \phi(y_{n-1}) - h_{n-1} &\leq \phi(y_{n-2}^n) - h_{n-2}.
\end{aligned} \tag{3.5}
$$

Without any loss of generality set $y_0^n = 0$. Since the functions $\phi(y), \varphi(y)$ and the holding cost rate are non-decreasing, $y_{n-2}^n \leq y_{n-1}^n$. Recursively applying (3.5) to the resulting inequality, we get (3.4).

From Lemma 1,

$$
\begin{aligned}
& y_{k+1}^m < \kappa && \text{for } 1 \leq k \leq m-1 \\
\Rightarrow\ & h_k - z(m) = \phi(y_k^m) - \min\left\{y_{k+1}^m, \kappa\right\} && \text{for } 1 \leq k \leq m-1.
\end{aligned} \tag{3.6}
$$

Case 1: There exists a solution to the set of $(m+1)$-optimality equations. As $z(m) \leq z(m+1)$ and the $y's$ are increasing in z, for $1 \leq k \leq m$, we have

$$
h_m - z(m) \geq h_m - z(m+1), \qquad y_k^m \leq y_k^{m+1}.
$$

So,

$$
\begin{aligned}
\Rightarrow \phi(y_m^m) - \kappa &\geq \phi(y_m^{m+1}) - y_{m+1}^{m+1} \geq \phi(y_m^m) - y_{m+1}^{m+1} \\
\Rightarrow y_{m+1}^{m+1} &\geq \kappa,
\end{aligned} \tag{3.7}
$$

from the m-optimality equations and the increasing nature of $\phi(y)$. Define $\delta \equiv z(m+1) - z(m)$, and $g_k := h_{m+1} - \delta$ for $k \geq m+1$. From these definitions, (3.7) and (2.13),

$$g_k - z(m) = \phi(y_{m+1}^{m+1}) - \min\left\{y_{m+1}^{m+1}, \kappa\right\} \qquad \text{for } k \geq m+1. \tag{3.8}$$

Consider the holding cost rates to be $(h_0, h_1, \cdots, h_{m-1}, h_m, g_{m+1}, \cdots)$, where beyond m the cost rates are set to g_{m+1} which is lower than the original holding costs. The optimality of the m-optimal policy under the new holding cost rates follows from the equations (2.4b), (3.6) and (3.8), using Theorem 1.

Case 2: A solution for the set of $(m+1)$-optimality equations does not exist. The following relation holds

$$\phi(y_{m+1}) - \kappa < h_{m+1} - z,$$

for any pair $(z, (y_1, \cdots, y_m, y_{m+1}))$ satisfying equations (2.4b) and (2.13) with $n = m+1, k \leq m$. This is also true when $z = z(m)$. Therefore,

$$\phi(\kappa) - \kappa < h_{m+1} - z(m). \tag{3.9}$$

Note that, in *Case 2*, due to the validity of the above inequality $\phi(\kappa)$ is finite. If $y_m^m \geq \kappa$, then the m-optimal policy satisfies the optimality equations for the modified holding cost rate truncated at m. This is similar to *Case 1*, where $g_k = h_m$ for $k \geq m+1$.

Assume $\kappa > y_m^m$. Consider a modified holding cost rate of $h'_{m+1} := z(m) + \phi(\kappa) - \kappa$ instead of h_{m+1}. For a system with the modified cost rate, $(z(m), (y_1^m, \cdots, y_m^m, \kappa))$ satisfies the $(m+1)$-optimality equations. Since $z(m)$

corresponds to the $(m+1)$-optimal policy for the modified system with lower cost rates, $z^*(m+1) \geq z(m)$, as in the assumptions.

Further using the arguments along the lines of *Case 1*, defining $g_k = h'_{m+1}$ for $k \geq m+1$, the optimality of the m-optimal policy for a system with smaller holding cost rates holds.

The result above imply $z^* \geq z(m)$, but since the resulting policy is an ergodic policy, $z^* \leq z(m)$. Hence the m-optimal policy is a optimal policy. The monotonicity of the optimal service rates follows directly from (3.4) and (2.14). □

The above proposition can be used to obtain an optimal policy only when a sequence of terminating optimal policies can be constructed. It is possible that no (z, y_1) pair satisfies the 1-optimality equations. In this case, arguing as in Theorem 4, *Case* 2 it is easy to see that the policy of just rejecting any customer is optimal.

3.2 Optimal Policy: Decreasing Sequence

If there exists a sequence of terminating optimal solutions, satisfying

$$z(n) > z(n+1) \quad \forall \quad n \geq 0, \tag{3.10}$$

then $z(n) \uparrow z(\infty)$ is the cost incurred by using the limiting control policy, i.e., $(\vec{\zeta}, \infty)$ where $\psi(y_m(z(\infty))) \uparrow \zeta_m$ for $m \geq 1$. From the discussion preceding Section 2.1.1 it is clear that the limiting service rate $\zeta_n < \infty$ for $n \geq 1$. Further, note that the limiting control policy is admissible, only if it is ergodic. To prove optimality of the limiting control policy we need to prove

$z(\infty) = z^*$ and ergodicity of the limiting policy. The next lemma proves the existence in a non-terminating policy of a system with a truncated holding cost rate $(h_0, h_1, \cdots, h_{n-1}, h_n, h_n, \cdots)$. This result is used in establishing the optimality of the limiting control policy.

Lemma 2. *Suppose the sequence of n-terminating optimal policies is decreasing, i.e., it satisfies (3.10). Then for each $n \geq 1$, there exists a sequence $(\hat{z}(n), (\hat{y}_1^n, \cdots, \hat{y}_n^n))$, such that,*

(i) The sequence satisfies (2.4b) and

$$\kappa > \hat{y}_k^n = \phi(\hat{y}_{k-1}^n) - h_{k-1} + \hat{z}(n) \quad \text{for } 2 \leq k \leq n \tag{3.11}$$

$$\kappa > \hat{y}_n^n = \phi(\hat{y}_n^n) - h_n + \hat{z}(n), \tag{3.12}$$

(ii) and the increasing nature of the $y's$ is preserved:

$$\hat{y}_1^n \leq \hat{y_2^n} \leq \cdots \leq \hat{y_n^n}. \tag{3.13}$$

Proof. As in [13], the proof is along the lines of construction of a sequence. Consider $(z, (y_1, \cdots, y_n))$ such that (2.4b) and the equality part in (3.11) are satisfied. Note that for each fixed z if a sequence of $y's$ exists it is unique. Define

$$\Delta_n(z) \equiv \phi(y_n(z)) - y_n(z) + z - h_n. \tag{3.14}$$

Let $S(n)$ represent the statement that there exists a z such that $\Delta_n(z) = 0$. If $\Delta_n(z(n)) \leq 0$, the n-optimal policy satisfies the optimality equations (2.4b)-(2.6). This is a contradiction as there exists a decreasing sequence (*in* *terms of cost*) of terminating policies. So,

$$\Delta_n(z(n)) > 0, \quad \forall \quad n \geq 1.$$

For $n = 1$, using the fact that the holding cost is increasing, we have

$$\Delta_1(h_0) = \phi(0) + h_0 - h_1 \leq 0.$$

From the construction of the sequence $y(z)$ as suggested in the beginning of the proof and definition (3.14), it follows that $\Delta_k(z)$ is an increasing continuous function in z. Therefore there exists a z such that $h_0 \leq z < z(1)$, i.e., $S(1)$ is true.

Assume $S(m)$ is true for $m \geq 1$. Note that $S(m)$ is true for some associated z, say $\hat{z}(m)$. Then the following holds,

$$\begin{aligned} &\Delta_{m+1}(z(m+1)) > 0, \\ &\quad \Delta_{m+1}(\hat{z}(m)) = \phi(y_{m+1}(\hat{z}(m)) - y_{m+1}(\hat{z}(m)) + \hat{z}(m) - h_{m+1} \qquad (3.15) \\ &\Rightarrow \Delta_{m+1}(\hat{z}(m)) = h_m - h_{m+1}, \end{aligned}$$

since, $S(m)$ is true and $y_{m+1}(\hat{z}(m)) = y_m(\hat{z}(m))$. Using (3.15) and the continuity of $\Delta_{m+1}(z)$, there exists a z say $\hat{z}(m+1)$, such that

$$\Delta_{m+1}(\hat{z}(m+1)) = 0, \quad \hat{z}(m) \leq \hat{z}(m+1) < z(m+1). \qquad (3.16)$$

Therefore the truth of statement $S(m+1)$ follows. Using induction $S(n)$ is true for all $n \geq 1$. Now consider $(\hat{z}(n), (\hat{y}_1^n, \hat{y}_2^n, \cdots))$ for some $n \geq 1$, such that (2.4b) and, the equality part of (3.11) and (3.12) are satisfied. Using inequality (3.16) for $m = n$, and Lemma 1,

$$y_k(\hat{z}(n)) < \kappa \quad \text{for } 1 \leq k \leq n. \tag{3.17}$$

As in Theorem 4, result (ii) is immediate. Then (i) follows from result (ii), (3.17), the construction of the sequence and the definition of $\hat{z}(n)$. $\square$

Lemma 3. *For a modified system with $(h_0, \cdots, h_{n-1}, h_n, h_n, \cdots)$ holding cost rate the following hold:*

(i) If $\hat{z}(n) \geq h_n$, the dynamic control problem is degenerate.

(ii) $(\vec{\eta}(n), \infty)$, where $\vec{\eta}(n) = \{\hat{\mu}_i^n\}$ is a optimal ergodic policy.

Proof. For clarity note that, elements of $\vec{\eta}(n)$ are dictated by $\hat{z}(n)$, and $\hat{\mu}_k^n = \hat{\mu}_n^n$ for $k \geq n$. The result follows from Proposition 4 in [13] and observing that $(\hat{z}(n), (\hat{y}_1^n, \cdots, \hat{y}_n^n))$ satisfies (2.4b), (3.11) and (3.12). $\square$

In the computation method suggested, if the optimal cost for a terminating system is strictly greater than the previous optimal cost, i.e, $z^*(n) > z^*(n+1)$, then Lemma 3 above result guarantees the existence of a non-terminating optimal policy for a system with holding cost truncated at h_n. Even if $z^*(n+1) \leq z^*(n+2)$, this result holds, i.e., the decreasing nature of the sequence of $z(n)$ is required only until *stage* $n+1$. The sequence of $\hat{z}(n)'s$ is non-decreasing, since the holding cost is non-decreasing. Also the $\hat{y}'s$ are bounded and increasing functions in terms of $\hat{z}$. From these properties and (3.13), considering the construction of non-terminating policies as in Lemma 2, we have

$$\begin{aligned} \hat{z}(n) \uparrow \hat{z}(\infty) \le z^* & \qquad \text{as } n \uparrow \infty, \\ \hat{y}_i^n \uparrow \ \hat{y}_i^* = \hat{y}_i(\hat{z}(\infty)) & \quad \text{for } i \ge 1 \\ \hat{y}_1^* \le \hat{y}_2^* \le \cdots . & \end{aligned} \tag{3.18}$$

Since $\psi(\cdot)$ is left-continuous and non-decreasing, (3.18) implies

$$\psi(\hat{y}_i^n) \uparrow \hat{\mu}_i^* = \psi(\hat{y}_i^*) \text{ as } n \uparrow \infty \quad \text{for each } i \ge 1. \tag{3.19}$$

If a optimal non-terminating policy described above is ergodic then using (2.2),

$$\begin{aligned} \hat{z}(n) &= \sum_{k=0}^{n-1} p_k(\vec{\eta}(n), \infty) \left\{ c(\hat{\mu}_k^n) + h_k \right\} + \sum_{k=n}^{\infty} p_k(\vec{\eta}(n), \infty) \left\{ c(\hat{\mu}_n^n) + h_n \right\} \\ \Rightarrow \hat{z}(n) &= \sum_{k=0}^{n-1} p_k(\vec{\eta}(n), \infty) \left\{ c(\hat{\mu}_k^n)) + h_k \right\} + M(n) \left\{ c(\hat{\mu}_n^n) + h_n \right\}, \end{aligned} \tag{3.20a}$$

where

$$M(n) = 1 - \sum_{k=0}^{n-1} p_k(\vec{\eta}(n), \infty) = p_n(\vec{\eta}(n), \infty) \frac{\hat{\mu}_n^n}{\hat{\mu}_n^n - 1}. \tag{3.20b}$$

Theorem 5. *If $h_n \uparrow h_\infty$ as $n \uparrow \infty$ and $z^* = \hat{z}(\infty) \ge h_\infty$, then the original problem is degenerate.*

Proof. The result follows from (3.18). □

Theorem 6. *If (3.10) holds and $\hat{z}(\infty) < h_\infty$, then the limiting control policy $(\vec{\mu}(\infty), \infty)$ is an ergodic optimal policy. This optimal policy is monotone in* *terms of number of customers in the system.*

Proof. For every ergodic policy $(\vec{\eta}(n), \infty)$ construct a policy $(\vec{\ddot{\mu}}(n), n)$, defined by

$$\ddot{\mu}_k^n = \begin{cases} \hat{\mu}_k^n & \text{if } 1 \leq k < n; \\ \hat{\mu}_k^n \frac{p_n(\vec{\eta}(n),\infty)}{M(n)} & \text{if } k = n. \end{cases}$$

By the definition of $z(n)$ and the constructed policy,

$$\begin{aligned} z(n) &\leq \hat{z}(n) + M(n)\left\{c(\ddot{\mu}_n^n) - c(\hat{\mu}_n^n) + \kappa\right\} \\ &\leq z^* + M(n)\kappa \\ &= z^* + p_n(\vec{\eta}(n), \infty)\frac{\hat{\mu}_n^n}{\hat{\mu}_n^n - 1}, \end{aligned} \tag{3.21}$$

where the second inequality holds since $\ddot{\mu}_n^n = \hat{\mu}_n(n)$. Since the degenerate case is excluded, (3.18) implies that, there exists N such that

$$\hat{z}(n) \leq \hat{z}(\infty) < h_n \quad \forall \quad n \geq N. \tag{3.22}$$

Therefore from (3.12), $\phi(\hat{y}_n^n) > \hat{y}_n^n$ for all $n \geq N$. This implies that $\psi(\hat{y}_n^n) > 1$ for all $n \geq N$, i.e.,

$$\begin{aligned} \hat{\mu}_n^n &> 1 \quad \forall \quad n \geq N, \\ \Rightarrow \psi(y_n^{n+1}) &> 1 \quad \forall \quad n \geq N \text{ and} \\ \hat{\mu}_n^m &> 1 \quad \forall \quad m \geq n \geq N. \end{aligned} \tag{3.23}$$

The implied inequality is immediate from the increasing nature of $y(z)$ and the non-decreasing nature of $\psi(y)$. From this point onwards unless specified $n \geq$

N. Using (2.2), and the structure of the non-terminating policies considered in Lemma 2, we have

$$p_n((\vec{\eta}(n), \infty)) \left\{ \frac{\hat{\mu}_n^n}{\hat{\mu}_n^n - 1} + \sum_{k=0}^{n-1} \left(\prod_{i=n}^{k+1} \frac{1}{\hat{\mu}_i^n} \right) \right\} = 1. \tag{3.24}$$

Note that in the case when $\hat{\mu}_i(n) = 0$ for some $i < N$, the equations above can be suitably modified. From (3.24) and (3.23)

$$p_n((\vec{\eta}(n), \infty) \downarrow 0 \text{ as } n \uparrow \infty. \tag{3.25}$$

Since any terminating policy is stationary and ergodic, and as decreasing sequences of terminating policies exists, $z(n) > z^*$. Therefore from (3.25), (3.21) and (3.18), (3.19) we have

$$z(n) \downarrow z^* \qquad \text{as } n \uparrow \infty \tag{3.26}$$

$$y_i^n \downarrow y_i^* = y_i(z(\infty)) \quad \text{for each } i \geq 1$$

$$y_1^* \leq y_2^* \leq \cdots \tag{3.27}$$

$$\psi(y_i^n) \downarrow \mu_i^* = \psi(y_i^*) \qquad \text{for each } i \geq 1. \tag{3.28}$$

The result follows immediately from (3.26), (3.27) and (3.28). □

Though $y_n^* < \kappa$ for $n \geq 1$, the optimal service rates associated with $z(\infty)$ can grow without bound as the number of customers in the system increases, i.e., the optimal service rates could be unbounded.

Chapter 4

Numerical Examples

The purpose of this chapter is to illustrate the algorithm with a few numerical examples. Before going into the details regarding the examples a result is presented. This result is helpful in obtaining a lower bound on the *optimality gap* the current control policy (after an iteration) and the optimal control policy.

Theorem 7. *If* $z(n) > z(n+1)$ *for some* $n \geq 1$, *and* $z(k) > z(n)$ *for* $k < n$ *then*

$$z(n) - z^* \leq \kappa - y_n^n. \tag{4.1}$$

Proof. Since $z(n) > z(n+1)$, the n-optimal policy is not an optimal policy. It follows that,

$$y_n^n < \kappa. \tag{4.2}$$

If the above inequality does not hold then setting $y_m = y_n^n$ for $m > n$, we have a solution pair to the optimality equations when the holding cost is modified to be $(h_0, h_1, \cdots, h_{n-1}, h_n, h_n, \cdots)$, from Theorem 1 and Lemma 1. This implies

$z^* = z(n)$ which is a contradiction. Set $h'_m = h_m - \theta_n$ for $m < n$, where $\theta_n = \kappa - y^n_n$. From (4.2) θ_n is positive. For a system with smaller modified holding costs $(h'_0, h'_1, \cdots, h'_{n-1}, h_n, h_n, \cdots)$, the pair $(z(n) - \theta_n, (y^1_n, y^2_n, \cdots, y^n_{n-1}, y^n_n, y^n_n, \cdots))$ satisfies the optimality equations. Hence

$$z(n) - \theta_n \leq z^*,$$

which immediately implies the result. $\square$

The result above helps us to terminate the algorithm after finite number of iterations when the gap is within an acceptable tolerance. Thus, we are able to achieve near optimal policies even when the optimal policy is the limiting policy. When the limiting policy is the optimal policy then, $\theta_n \downarrow 0$ as $n \uparrow \infty$.

It is interesting that Theorem 7 holds irrespective of the holding cost structure. In the following numerical examples the holding cost structure is $h_n = h_0 + s(n - M + 1)^+$. In the first example we consider a similar cost for service as in [13], i.e., $c(x) = x^2$. Figure 4.1 shows the variation in the optimal buffer size as the multiplicative factor s in the holding cost and rejection cost are varied. It is clear that for fixed κ, the optimal buffer size increases with decreasing s and further, this effect is more prominent for lower values of s. An interesting result is the apparent absence of a monotone a trend in optimal buffer size with respect to κ, for a fixed value of s. However, the behavior (increasing or decreasing) of optimal buffers size with respect to κ, is not the same across s. We can also observe that this apparent lack of trend is prominent only once the optimal buffer size 'saturates'. As expected, the optimal buffer size is high when the rejection cost is high and s is low. For low values of the rejection cost, irrespective of s, all customers are rejected.

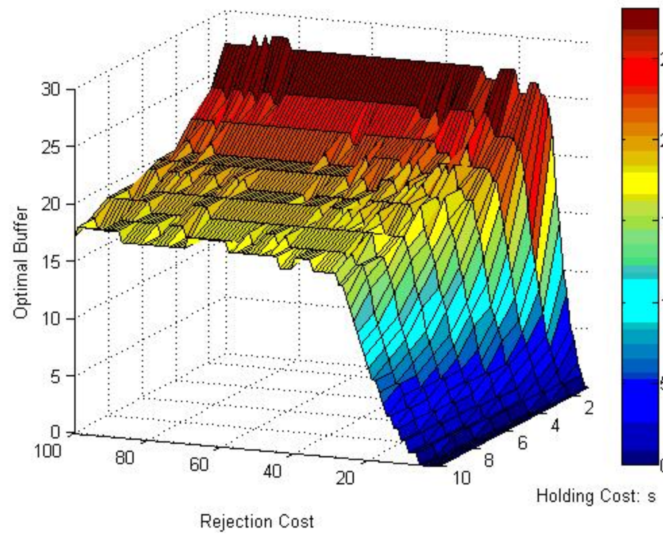


Figure 4.1: Optimal buffer size: $c(x) = x^2$

This is due to the chosen values of $h_0 = 10$ and $M = 1$. The optimal cost is plotted against s and κ in Figure 4.2. Clearly for a fixed κ, the optimal cost increases with increasing s. Unlike in the case of the optimal buffer size, a clear trend is present. Optimal cost increases with increasing κ, for a fixed value of s. However, note that the optimal values saturate relatively quickly and this saturation occurs faster for lower values of s. The optimal buffer size and optimal objective values have opposite trends with respect to s, i.e., as the optimal cost decreases the optimal buffer size increases. For smaller values of s and with a negligible increase in optimal cost, large optimal buffer size can be achieved.

The next example is selected such that the cost for service per unit time approaches the rejection cost as the service rate approaches ∞. In this case **A5** holds with equality. The cost for service is $c(x) = x - x^{\frac{1}{1+\varepsilon}}$. The cost function depends on the parameter ε. In Figures 4.3 and 4.4, the optimal buffer size

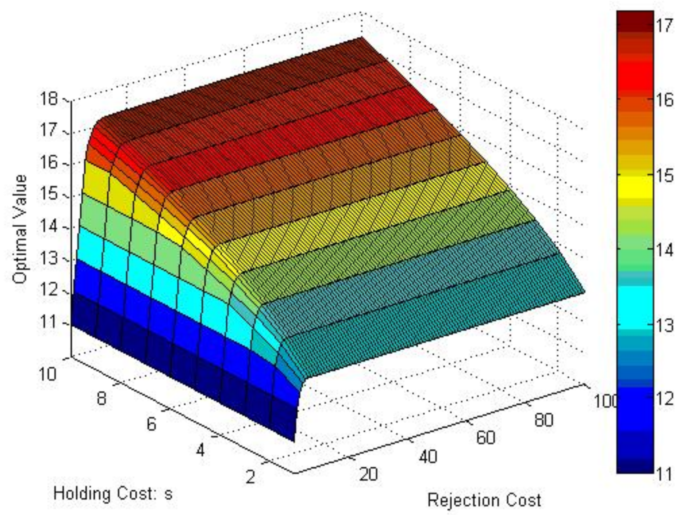


Figure 4.2: Optimal cost: $c(x) = x^2$

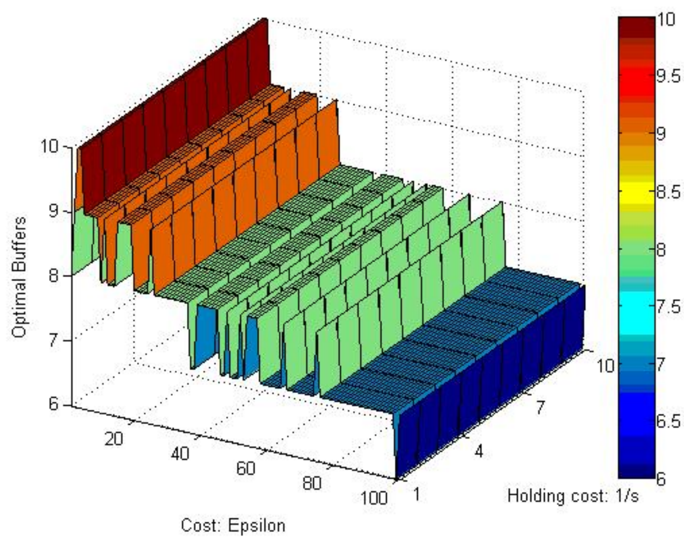


Figure 4.3: Optimal buffer size: Example 2

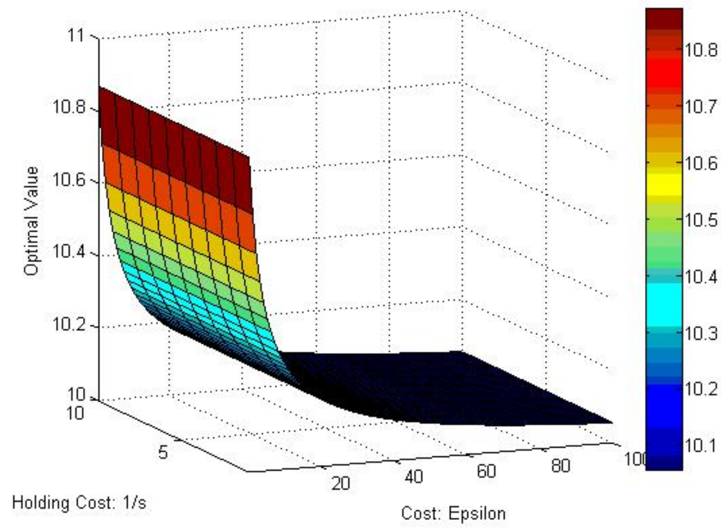


Figure 4.4: Optimal cost: Example 2

and optimal cost, respectively, are plotted on the vertical axis, against ε and $\frac{1}{s}$. The other parameters are fixed at $\kappa = 1, M = 1$ and $h_0 = 1$. Similar to the first example there is an apparent lack of a trend with respect to ε when the value of s is fixed. Note that both the optimal cost and optimal buffer size are insensitive to the value of s. Clearly, as the cost function becomes smaller, i.e., ε increases, optimal cost decreases. The behavior in this example is somewhat unexpected.

The computation methodology suggested in this work is efficient in the following sense: in each iteration the search for optimal value is in terms of a single variable, namely z. For a fixed value of threshold it does a search for an optimal value of z to satisfy the terminating optimality equations. This search being in single dimension requires less computation. If the computation of $\phi(\cdot)$ is not intense, like in case of polynomial or well behaved cost functions, then the algorithm is expected to perform well. For the above examples for a given instance that is chosen value of parameter the algorithm took around 30 seconds on an average to terminate, which is fast. The problem instances were solved in Matlab, on Linux systems with two 1.8GHz Pentium Xeon processors and 1GB Ram.

Chapter 5

Job Shop Model

A job shop consists of multiple stations and multiple job types, each with a number of jobs. Further, each station might have multiple servers. Each job of a particular type has to complete certain processing steps in a specific order, i.e., follow a fixed route through the job shop. Associated with each step is a station and the step can be completed at any one of the servers associated with a particular station. Below the notation and terms to be used throughout this work are introduced.

A job shop is comprised of a set of job types $\Phi := \{1, \ldots, J\}$, which are processed at stations belonging to the set $\Psi := \{1, \ldots, \mathcal{I}\}$. Each station has a set of servers, say $\varpi(i) := \{1, \ldots, M(i)\}$ for station i. A fixed route $1, \ldots, K(j)$, is the particular sequence of operations associated with job type j. For job type j, step $k \in \{1, \ldots, K(j)\} := \Upsilon(j)$ is done at one of the servers of station $\rho(j,k) \in \Psi$, and the processing time is $m(j,k)$. Let $\mathbf{H}$ be the set of all job classes, i.e., $\mathbf{H} := \{(j,k) : j \in \Phi, k \in \Upsilon(j)\}$. The number of jobs present initially at time $t = t_0$ is $a(j,k)$, and the total initial number of jobs associated with job class (j,k) is $b(j,k) := \sum_{s=1}^{k} a(j,s)$. Let $\sigma(i)$ denote the set of job classes or (j,k) pairs being processed at station i.

When any of the above parameters describing the job shop, such as mean processing time are not know with certainty, the superscript ω is used. ω denotes a scenario in the space of scenarios Ω. This usage is illustrated below. In the job shop model studied in this work, the number of jobs present at time $t = t_0$ is uncertain. To indicate the stochastic element, the nomenclature for the initial number of jobs of class (j,k) is $a^\omega(j,k)$, for scenario ω. The job shop can be represented as $\mathbf{J} := \{\varpi, \Upsilon, \rho, a\}$, where $a := \{a^\omega(j,k) : \omega \in \Omega, (j,k) \in \mathbf{H}\}$ and so on. We use the following to denote the maximum and minimum, $\dot{m}^\omega := \max_{(j,k)\in\mathbf{H}} m^\omega(j,k)$ and $\ddot{m}^\omega := \min_{(j,k)\in\mathbf{H}} m^\omega(j,k)$, where the notation is used for other parameters too as appropriate. The minimum is taken only over the set of non-zero quantities. In the notation we use for the quantities in the jobs shop, entities like job type $j \in \Phi$, as well as time t are used as functional arguments. For example, associating $x(\cdot)$ with allocation process, $x^\omega(j,k,l,t)$ indicates whether or not at time $t \geq 0$ server $l \in \varpi(\rho(j,k))$ is processing a job of job class $(j,k) \in \mathbf{H}$, in scenario $\omega \in \Omega$.

We denote the expected makespan under a policy π as $T(\pi)$. Policy π is the process of allocating time to different job classes at all servers. For other quantities, to keep the notation less cumbersome, the dependence on policy is not represented. It is assumed that in each context where the quantities appear the dependence is implicit and well defined. For example, the expected makespan for the job shop (all job types) under scenario ω is T^ω. Similarly, let $t^\omega(j)$ be the makespan of job type (j) in scenario ω. From the above discussion, for a policy π, we have the following definitions

$$T(\pi) := \mathbf{E}_{\Omega}\left[T^{\omega}\right] \tag{5.1a}$$

$$T^{\omega} := \max_{j \in \Phi}\left\{t^{\omega}(j)\right\} \qquad \text{for } \omega \in \Omega. \tag{5.1b}$$

The expectation in (5.1a) is with respect to the probability measure $\mathbf{P}$ over the scenario space Ω and hence the subscript.

We now introduce the notation used when the processing times are doubly stochastic. In our basic model with parameter uncertainty, for each fixed scenario $\omega \in \Omega$, all quantities associated with the job shop are deterministic. In the doubly stochastic model, for each fixed scenario $\omega \in \Omega$, certain quantities, such as processing times, are allowed to be random. Hence, each ω induces a probability measure and related functionals, such as expectation. In most cases, we explicitly denote the dependence of the scenario induced probability measures and expectation by $\mathbf{P}_{\omega}$ and $\mathbf{E}_{\omega}$, respectively. Further, to denote the random nature of the parameters we use a bar. For example, $\bar{m}^{\omega}(j,k)$ is a random variable representing the processing time of class $(j,k) \in \mathbf{H}$ under scenario $\omega \in \Omega$. To account for randomness in processing times, the definition of T^{ω} is modified as follows

$$T^{\omega} := \mathbf{E}_{\omega}\left[\bar{T}^{\omega}\right] \qquad \text{for } \omega \in \Omega. \tag{5.2}$$

The expectation here is with respect to the scenario induced probability measure $\mathbf{P}_{\omega}$. Combining (5.1a) and (5.2), we have the following generic definition of expected makespan under a policy π,

$$T(\pi) = \mathbf{E}_{\Omega}\left[\mathbf{E}_{\omega}\left[\bar{T}^{\omega}\right]\right].$$

While the continuous job shop model is a relaxed version of the discrete job shop model, the fluid model is obtained by considering mean processing times along with certain modifications in the continuous job shop model. A tilde is used to indicate the quantities corresponding to the fluid model. For example $\tilde{T}^{\omega}$ is the makespan in the fluid model under scenario $\omega \in \Omega$. An underline indicates that the quantity is related to the continuous job shop. Similarly, for quantities in the continuous job shop model with random processing times, a bar is used. For example, the random variable $\bar{t}^{\omega}(j)$ is the makespan for job type $j \in \Phi$ under scenario $\omega \in \Omega$ in the discrete job shop model. $\mathbf{T}$ is the optimal expected drain time of the fluid model. Similarly all qualities corresponding to optimal solution of the fluid model are boldfaced.

We assume the following:

a1. The structure of the job shop $(\varpi, \Upsilon, \rho, a)$ and, $\mathbf{P}$ and $\mathbf{P}_{\omega}$ for all job classes are known to the controller.

a2. Average work to be done for a job class (j, k), for some scenario $\omega \in \Omega$ satisfies $b^{\omega}(j,k)m^{\omega}(j,k) > 0$.

a3. Processing times are uniformly bounded in each scenario, i.e., $\bar{m}^{\omega}(j,k) \leq \mathbf{U}$, for some $\mathbf{U} > 0$, for all $(j,k) \in \mathbf{H}$ in each scenario $\omega \in \Omega$.

a4. Initial number of jobs is bounded.

a5. The number of scenarios is finite, $|\Omega| < \infty$, and probability of occurrence of each scenario $\omega \in \Omega$ is non-zero, $\mathbf{P}_\Omega(\omega) > 0$.

The rest of the chapter includes discussion regarding various policies, fluid and continuous job shop models, and asymptotic optimality. The organization is as follows. In section 5.1 we formulate the generic job shop scheduling problem for makespan objective. In this formulation processing times are considered to be non-random. We introduce the set of cyclic policies in next section 5.2. The constraints associated with cyclic policies and different versions of the policy are discussed. In section 5.3, under a specific set of policies a relaxed version of the job shop, the continuous job shop model, is derived by relaxing certain constraints. Further, the continuous job shop model is modified to obtain the fluid model and the fluid model is shown to provide lower bound on the optimal exceeded makespan for the continuous job shop model. Finally in section 5.4 different scalings studied in the work are introduced along with the notion of asymptotic optimality.

5.1 Problem Formulation

The scheduling policy used by the controller, along with the time evolving randomness governs the way in which the job shop changes with time, i.e., the dynamics of a job shop. In the present model, time evolving randomness is absent. Hence, for each scenario the dynamics and in turn the resulting value of the objective function, in this case makespan, are completely governed by the scheduling policy. A scheduling policy is a mapping based on time and the job shop state onto the set of feasible actions. The set of feasible actions is determined to some extent by nature of the job shop. For example, in a (discrete) job shop only one job can be processed by a server at a given time.

Based on specific restrictions in place more constraints are added, reducing the space of feasible policies. In this section the makespan minimization problem is formulated for a discrete job shop under very generic setting. Additional constraints are added later in this work as specific model settings are developed. The constraints in the formulation restrict the scheduling policies to allow feasible job shop dynamics. Since the feasible space is the same irrespective of the objective function, the makespan objective can be replaced by any other objective function such as general holding cost.

The generic scheduling problem over a class of policies $\mathcal{F}$, can be formulated as below, where $X^{\omega}(\cdot, t)$ are counting processes. In particular $X^{\omega}(j, k, l, t)$ is the number of jobs of type j which have been through step k at the l^{th} server of station $\rho(j, k)$ up till time $t \geq 0$, under scenario ω. The objective is to minimize the expected value of makespan, T^{ω}. The problem formulation (**GJS**) is as follows:

$$\min_{\pi\in F}\{\mathbf{E}_\Omega[T^\omega]\} \tag{5.3a}$$

$$\text{s.t. } T^\omega \geq t^\omega(j) \tag{5.3b}$$

$$X^\omega(j,k,t)=\sum_{l\in\varpi(\rho(j,k))}X^\omega(j,k,l,t) \tag{5.3c}$$

$$\sum_{(j,k)\in\sigma(i)}x^\omega(j,k,l,t)\leq 1 \tag{5.3d}$$

$$m^\omega(j,k)X^\omega(j,k,l,t)\leq\int_0^t x^\omega(j,k,l,s)ds \tag{5.3e}$$

$$X^\omega(j,K(j),t^\omega(j))=b^\omega(j,K(j)) \tag{5.3f}$$

$$[X^\omega(j,k-1,t)+a^\omega(j,k)]\,m^\omega(j,k)\geq\sum_{l\in\varpi(\rho(j,k))}\int_0^t x^\omega(j,k,l,s)ds, \tag{5.3g}$$

$$X^\omega(j,0,t)=0$$

$$x^\omega(j,k,l,t)\in\{0,1\}$$

$$t\geq 0, i\in\Psi, j\in\Phi$$

$$(j,k)\in\mathbf{H}, l\in\varpi(\rho(j,k))$$

The expectation in the objective function is taken when the job shop is operating under a for a policy π. An additional constraint, denoted by pr indicates that a policy is preemptive non-resume, i.e., jobs may be preempted by certain events like breakdowns or higher priority jobs during processing. Further under such a policy, when processing resumes on a preempted job, the processing must resume anew. This cannot be expressed easily in analytic form. The constraints in the above formulation along with pr hold for any feasible policy, i.e., the policy space defined by these constraints is that of all feasible policies. This formulation allows the class of policies to depend on the state information of the job shop. There might be additional constraints which are inherent in the definition of the policy space $\mathcal{F}$ under consideration.

The explanation for some of the above equations is given here. In (5.3c), $X^{\omega}(j,k,t)$ captures the total number of jobs of a particular job class completed till time t. Selection of a particular job class (j,k) to be processed at server l, is captured by $x^{\omega}(j,k,l,t)$. The constraint that a server can process only one job at any given time, is modeled as (5.3d). A job cannot be released at a step unless it is processed at all the previous steps including the present step which is modeled in (5.3g). The constraint (5.3e), on the number of completed jobs at a server states that the total time spent for processing the jobs must be less than that allocated for the step.

Equation (5.3f) keeps track of the time when all the jobs of a particular type are completed. Completion of the final step in the flow is an indication that processing of the jobs is complete. Makespan under a particular scenario is the maximum over all the completion times of the job types, which is captured by (5.3b). Observe that this constraint can be rewritten for a given scenario $\omega \in \Omega$, in the form of the following relation:

$$T^{\omega} = \max_{j \in \Phi} \left\{ t^{\omega}(j) \right\}. \tag{5.4}$$

5.2 Policy Set

It might be desirable under certain circumstances to have a fixed *non-adaptive* policy, for example, when information about the complete state of the job shop is not available. The implementing entity may only process the jobs in some predefined manner which does not depend on the realization of the parameters.

The controller knows the total time allocated to a given job class at a server, up till a given time, i.e., the left side of the inequality (5.3e). Such data and, data known with certainty like processing route of jobs and number of servers, are called *observable*. Since the controller can keep track of total time allocated to each class at each server, this data is considered observable. All other data called *unobservable*. In the present formulation the policy is not restricted to using only observable data, in the sense mentioned above. From now on we consider *non-adaptive* policies, those that are not functions of unobservable data.

5.2.1 Cyclic Policy

A specific set of non-adaptive policies are introduced in this section. This class of policies, which are time *independent* are simpler and easier to implement; as lesser tracking is needed. In later chapters classes of modified cyclic policies are introduced. All these policy classes are different variations of cyclic policy mentioned in Chapter 1. These policy sets are specified through addition or relaxation of constraints in the job shop formulation presented earlier.

The first class of such policies studied in this work are called *fixed allocation* cyclic policies ($\mathcal{F}_{\mathbf{f}}$). These policies are described below. Similar to cyclic policies for traffic light control, the allocation of time to different classes is in terms of cycles. The basic unit of such a policy is a set of cycles $C(l,i)$ for each server $l \in \varpi(i)$ and station $i \in \Psi$. A fixed fraction of the cycle, $z(l,j,k)$, is allocated to job class (j,k), at server $l \in \varpi(\rho(j,k))$. The time allocated to a job class in a cycle is in a single *chunk*, as in the case of traffic light control. Rephrased, if in a cycle a job class is selected for processing, irrespective of the presence of jobs, no other class is processed at the server unless the time allocated to the job class is complete. Thus, idle time is induced in the absence of the jobs. If the time allocated to a job class in a cycle is not a multiple of the processing time then there is a possibility for preemption, resulting in additional processing time. This effect can be more prominent when the processing times are random. A complete cycle at a server l of station i, $C(l,i)$, might include additional idle time $I(l,i)$. In accordance with the initial model **GJS**, restricting to this class of policies adds the following set of constraints (**CYC**):

$$\int\limits_{nC(l,\rho(j,k))}^{(n+1)C(l,\rho(j,k))} x^{\omega}(j,k,l,s)ds \leq z(l,j,k) \tag{5.5a}$$

$$\int\limits_{nC(l,\rho(j,k))}^{(n+1)C(l,\rho(j,k))} x^{\omega}(j,k,l,s)ds = \int\limits_{t}^{t+z(l,j,k)} x^{\omega}(j,k,l,s)ds \tag{5.5b}$$

$$I(l,i) + \sum_{(j,k)\in\sigma(i)} z(l,j,k) = C(l,i) \tag{5.5c}$$

$$nC(l,i) \leq t \leq (n+1)C(l,i)$$

$$z(l,j,k), C(l,i) > 0, I(l,i) \geq 0$$

$$n \in \mathbb{N}, i \in \Psi(j,k) \in \mathbf{H}, l \in \varpi(i), \omega \in \Omega.$$

Note that these constraints allow job classes to be processed in any sequence within a cycle. However, constraint (5.5b) forces the time allocated to be in a single chunk, as described previously. The pair that define a policy $\pi_f \in \mathcal{F}_f$ are:

1. time allocations $\mathbf{z} := \{z(j,k,l) : (j,k) \in \mathbf{H}, l \in \varpi(\rho(j,k))\}$,

2. cycle length $\mathbf{C} := \{C(l,i) : l \in \varpi(i), i \in \Psi\}$.

Bounds on the expected makespan under the policy can be obtained in terms of these parameters. However, fixing these parameters does not completely determine the manner in which the job shop state evolves. There still exists flexibility in a cycle in the manner in which time is allocated to each job class.

Once a cycle is started, irrespective of the presence of jobs, the server or station is considered to be utilized till the cycle is complete. As a result, even if all jobs to be processed at the server are completed before completion of the cycle, the whole cycle is counted towards calculation of the makespan. The rational for this constraint is the inability of the controller to predict if the complete system is empty. However, he is assumed to know information regarding the presence of jobs at a server. The traffic light example provides some motivation for this assumption. However, in the case of a single station with no reentry this assumption might appear a bit artificial. For convenience we have not formulated this constraint in terms of mathematical relations. Instead, we treat this just as another constraint like *pr* and denote it by *ccy* (short for complete cycle).

A further restricted class of policies called *fixed interval* policies ($\mathcal{F}_{\mathbf{fi}}$) is defined. The additional constraint is:

C1. Jobs arriving after the beginning of a cycle are not processed within the same cycle.

Such a policy is called a *gated* policy in the queueing literature on polling stations. The policies can be viewed as processing jobs available at the beginning of the cycle, up to a fixed number defined by the allocated time. Note that the additional constraint assumes that the controller posses knowledge of certain unobservable data, namely which jobs were processed in the present cycle and which were completed in a previous cycle. Consider the following surrogate model, in which an intermediate buffer is present where jobs processed in a given cycle are stored, until the present cycle at the next station is complete. In this model the unobservable nature of the policies is preserved. A similar policy was introduced and shown to be asymptotically optimal in [4]. For each job class, the heuristic the authors suggest allocates an integral multiple of the processing time at all the associated servers. Also, the cycle lengths are same across all servers and only jobs present at the beginning of cycle are processed during a cycle.

Consider a further restriction:

C2. In a cycle, a maximum of $\gamma^{\omega}(j,k)$ jobs of job class $(j,k) \in \mathbf{H}$, can be processed at server $l \in \varpi(\rho(j,k))$.

Under this additional constraint, the resulting class of policies is called *fixed number* cyclic policies ($\digamma_{fn}$). Restriction **C2** can be scenario dependent and hence the policy maker is assumed to have additional information, namely she observes which scenario is realized. If $\pi_{fn} \in \digamma_{fn}, \pi_{fi} \in \digamma_{fi}$ and $\pi_f \in \digamma_f$ are policies with the same parameters $\{\mathbf{z}, \mathbf{C}\}$ then due to structure of the polices the following monotone relation holds,

$$T(\pi_{fn}) \geq T(\pi_{fi}) \geq T(\pi_f). \tag{5.6}$$

In fact similar relation holds for the expected makespan under each scenario.

5.3 Continuous Models

We introduce the fluid and continuous job shop models in this subsection. In this section we assume that the processing times though scenario dependent are non-random. The job shop scheduling problem for makespan minimization under cyclic policies is a stochastic integer program, which is tough to solve. In fact the recognition version of the deterministic two machine job shop makespan problem is itself strongly NP-complete, see [17] by Hall et al. Hence, the next best thing to solving the problem is to develop near optimal heuristics. The present formulation does not provide much insight regarding how to develop a good heuristic. For this purpose we relax various constraints to develop a 'continuous job shop' model where all jobs of a particular type are combined into a single fluid.

5.3.1 Continuous Job Shop

Consider the the following relaxations to the discrete job shop model defined by model GJS, constraint set CCY and, constraints *pr* and *ccy*:

R1 Relax the counting processes to be stochastic non-deceasing processes and drop constraint *pr*. Hence, partial job processing is allowed.

R2 Relax integrality constraint (5.3d). This allows simultaneous server allocation among multi-job classes.

R3 Combine all servers at a station into a single server. This implies the index l and constraint (5.3c) are dropped.

R4 Drop constraint (5.5b). So, processing of fluid can be spread across the whole cycle.

The resulting model due to relaxation (**R1**)-(**R4**) is called continuous job shop. In continuous job shop we denote the quantities with a underline. In a discrete job shop it is apparent that we can safely restrict cycle lengths to be greater than $\max_{\omega \in \Omega} \{\dot{m}^{\omega}\}$. However, a similar restriction is not present in the above described relaxed model. The cycle lengths can be as close to 0 as possible. In the below described fluid model we can view the cycle lengths to be instantaneous, i.e., set to 0. Let $\underline{F}_f$ be the relaxed version of fixed allocation cyclic policy associated with the continuous job shop. In this work, use of underline with other policy sets has similar meaning. Combining this observation along with the relaxation (**R1**)-(**R4**), we obtain the following formulation (CJS) for the continuous job shop model:

$$\min_{\underline{\pi}_f \in \underline{F}_f} \left\{ \mathbf{E}_\omega \left[\underline{T}^\omega \right] \right\} \tag{5.7a}$$

$$\text{s.t. } \underline{T}^\omega \geq \underline{t}^\omega(j) \tag{5.7b}$$

$$\sum_{(j,k)\in\sigma(i)} x^\omega(j,k,t) \leq M(\rho(j,k)) \tag{5.7c}$$

$$m^\omega(j,k)\underline{X}^\omega(j,k,t) = \int_0^t \underline{x}^\omega(j,k,s)ds \tag{5.7d}$$

$$\underline{X}^\omega(j,K(j),\underline{t}^\omega(j)) = b^\omega(j,K(j)) \tag{5.7e}$$

$$\left[\underline{X}^\omega(j,k-1,t) + a^\omega(j,k)\right] m^\omega(j,k) \geq \sum \int_0^t \underline{x}^\omega(j,k,s)ds \tag{5.7f}$$

$$\int_{n\underline{C}(\rho(j,k))}^{(n+1)\underline{C}(\rho(j,k))} \underline{x}^\omega(j,k,s)ds \leq \underline{z}(j,k) \tag{5.7g}$$

$$\int_{n\underline{C}(\rho(j,k))}^{(n+1)\underline{C}(\rho(j,k))} \underline{x}^\omega(j,k,s)ds = \int_t^{t+\underline{z}(j,k)} \underline{x}^\omega(j,k,l,s)ds \tag{5.7h}$$

$$\sum_{(j,k)\in\sigma(i)} \underline{z}(j,k) = \underline{C}(i) \tag{5.7i}$$

$$n\underline{C}(i) \leq t \leq (n+1)\underline{C}(i)$$

$$\underline{X}^\omega(j,0,t) = 0$$

$$\underline{z}(j,k), \underline{C}(i) > 0$$

$$n \in \mathbb{N}, i \in \Psi(j,k) \in \mathbf{H}, \omega \in \Omega.$$

Note that (5.7e) holds with equality due to **R1**. In this continuous job shop model (CJS), $\underline{z}(j,k)$ can be viewed as the maximum capacity allocated to job class (j,k) in a cycle. For a job class (j,k), assuming that all the initial jobs are available when needed without any delay, a minimum of $\left\lceil \frac{b^\omega(j,k)m^\omega(j,k)}{\underline{z}(j,k)} \right\rceil$ cycles is required to complete all jobs. From this observation we derive the following lemma.

Lemma 4. *For a continuous job shop operated under a policy $\underline{\pi}_f \in \digamma_f$ with parameters $\{\underline{z}(j,k) : (j,k) \in \mathbf{H}\}$ and $\{\underline{C}(i) : l \in \varpi(i), i \in \Psi\}$, the makespan under each $\omega \in \Omega$ satisfies*

$$\underline{T}^\omega \geq \max_{j \in \Phi} \left\{ \left\lceil \frac{b^\omega(j,k)m^\omega(j,k)}{\underline{z}^\omega(j,k)} \right\rceil C(\rho(j,k) \right\}. \tag{5.8}$$

Proof. From the definition of cycle for any $n \in \mathbb{N}$, we have

$$\underline{X}^\omega(j,k,nC(\rho(j,k))) \leq n \frac{\underline{z}(j,k)}{m^\omega(j,k)}. \tag{5.9}$$

Also, summing (5.7f) across job classes and (5.7d) we get

$$b^\omega(j,k)m^\omega(j,k) \geq \underline{X}^\omega(j,k,nC(\rho(j,k))). \tag{5.10}$$

If $n^{'}$ is the cycle by when all jobs of class (j,k) are completed, then (5.10) holds with equality. Hence, for cycle $n^{'}$, combining (5.9) and (5.10) we have

$$\begin{aligned} n^{'} \frac{\underline{z}(j,k)}{m^\omega(j,k)} &\geq b^\omega(j,k) \\ \Rightarrow n^{'} &\geq \frac{b^\omega(j,k)m^\omega(j,k)}{\underline{z}(j,k)}. \end{aligned} \tag{5.11}$$

Even if jobs of type j are complete by some cycle $p \in \mathbb{N}$ at station $\rho(j, K(j))$, but the time for completion of p cycles $pC(\rho(j, K(j)) \leq n^{'}C(j,k)$, then due to constraint *ccy*, $\underline{t}^{\omega}(j) = n^{'}C(j,k)$. Hence, from (5.11) the result (5.8) follows.

□

5.3.2 Fluid Model

We modify the continuous job shop model as follows:

R4 Drop constraints *ccy* and 5.7h.

R5 Restrict the amount of capacity allocated to a job class at each instance $t \geq 0$ for each $(j,k) \in \mathbf{H}$ under each scenario $\omega \in \Omega$, $\underline{x}^{\omega}(j,k,t) \leq \frac{z(j,k)}{C(\rho(j,k))}$.

Modification **R5** is a restriction and is similar to having instantaneous cycles. Due to the the restriction, it is not immediately evident why this would lead to a model which provides a lower bound on the optimal makespan of the continuous job shop model. The modified model described by equations (**R4**)–**R5**) and (5.7a)–(5.7i) is called fluid model. Jobs are processed in a continuous fashion like flow of fluid through a system, hence the name. Considering percentage allocation at each instance of time instead of in a cycle, the fluid model can be re-formulated as below:

$$\min_{\mathbf{u}} \left\{ \mathbf{E}\left[\tilde{T}^{\omega}\right] \right\} \tag{5.12a}$$

$$\text{s.t.} \quad \tilde{T}^{\omega} \geq \tilde{t}^{\omega}(j) \tag{5.12b}$$

$$u(j,k) \geq 0$$

$$\sum_{(j,k)\in\sigma(i)} u(j,k) \leq 1 \tag{5.12c}$$

$$\tilde{x}^{\omega}(j,k,t) \leq u(j,k)M(\rho(j,k)) \tag{5.12d}$$

$$m(j,k)\tilde{X}^{\omega}(j,k,t) \leq \int_0^t \tilde{x}^{\omega}(j,k,s)ds \tag{5.12e}$$

$$\tilde{X}^{\omega}(j,K(j),\tilde{t}(j,\omega)) \geq b^{\omega}(j,K(j)) \tag{5.12f}$$

$$\left[\tilde{X}^{\omega}(j,k-1,t) + a^{\omega}(j,k)\right] m^{\omega}(j,k) \geq \int_0^t \tilde{x}^{\omega}(j,k,s)ds, \tag{5.12g}$$

$$X^{\omega}(j,0,t) = 0$$

$$t \geq 0, i \in \Psi, j \in \Phi$$

$$(j,k) \in \mathbf{H},$$

where $u(j,k)M(\rho(j,k))$ is capacity of the station allocated to a job class at any instance, $\tilde{X}(\cdot)$ is continuous process representing the total amount of fluid processed (jobs completed) and $\tilde{x}(\cdot)$ is the capacity actually used for processing a job class at a given instance. Note that in the above formulation too like previous formulations GJS and CJS, there is flexibility not use the total capacity available whenever possible. However, based on the problem structure, we can safely conclude that there is no advantage in not utilizing the total available capacity whenever possible. So, in the following discussion we restrict our consideration to such policies.

We define $\tilde{t}^{\omega}(j,k)$ to be the amount of time taken to empty a buffer. Then from the structure of policies under consideration and (5.12g), we get

$$\tilde{t}^{\omega}(j,1) = \frac{a^{\omega}(j,1)m^{\omega}(j,1)}{u(j,1)}$$
$$\Rightarrow b^{\omega}(j,k) = \begin{cases} \frac{u(j,2)M(\rho(j,2))}{m^{\omega}(j,2)}\tilde{t}^{\omega}(j,1) & \text{if } \frac{\tilde{t}^{\omega}(j,1)}{\tilde{t}^{\omega}(j,2)} \leq 1, \\ \frac{u(j,2)M(\rho(j,2))}{m^{\omega}(j,2)}\tilde{t}^{\omega}(j,2) + \left[\tilde{t}^{\omega}(j,1) - \tilde{t}^{\omega}(j,2)\right]\frac{u(j,1)M(\rho(j,1))}{m^{\omega}(j,1)} & \text{o.w.} \end{cases}$$

Extending the above arguments to all job classes, we get the below mentioned explicit expression for makespan and representation of the makespan problem (FS):

$$\min_{\mathbf{u}} \left\{ \mathbf{E}\left[\tilde{T}^{\omega}\right] \right\} \tag{5.13a}$$
$$\text{s.t.} \quad \tilde{T}(\omega) = \max_{(j,k)\in\mathbf{H}} \left\{ \frac{m^{\omega}(j,k)b^{\omega}(j,k)}{u(j,k)M(\rho(j,k))} \right\} \tag{5.13b}$$
$$\sum_{(j,k)\in\sigma(i)} u(j,k) \leq 1 \tag{5.13c}$$
$$u(j) > 0$$
$$(j,k) \in \mathbf{H}, \omega \in \Omega.$$

This stochastic program is guaranteed to have a optimal fluid solution, while it is is not clear why a optimal solution would exists in case of continuous job shop model. The optimal makespan is $\mathbf{T}$ and an optimal solution (allocation) is $\{\mathbf{u}(j,k) : (j,k) \in \mathbf{H}\}$. From formulation FS we readily have the following lemma,

Lemma 5. *If $\underline{T}^{\omega}$ is the makespan in scenario $\omega \in \Omega$ on implementing policy*

$\underline{\pi}_f$ *in a specific manner, and* $\tilde{T}^\omega$ *is the fluid makespan under policy* $\tilde{\pi}_f$ *defined as:*

$$\tilde{x}^\omega(j,k,t) = \begin{cases} \frac{z(j,k)}{\underline{C}(\rho(j,k))} & \textit{if } \tilde{X}^\omega(j,k,t) > 0 \\ \tilde{x}^\omega(j,k-1,t) & \textit{if } \tilde{X}^\omega(j,k,t) = 0, k > 1 \\ 0 & o.w, \end{cases}$$

then $\underline{T}^\omega \geq \tilde{T}^\omega$.

So, from Lemma 5 we have that the optimal expected makespan for FS provides a lower bound on optimal expected makespan for CJS.

5.4 Asymptotic Optimality and Scaling

Consider a sequence of job shops $\{\mathbf{J_1}, \mathbf{J_2}, \cdots\}$. For this sequence of job shops a scaling defines how in the scaled job shop $\mathbf{J_n}$ the number of machines and number of jobs are dependent on the scaling parameter n. Two interesting scalings which have been studied in literature are the fluid scaling and Halfin-Whitt scaling, see [16, 21]. In the literature, fluid, diffusion, and other scalings are often performed with respect to the initial number of jobs in the system as well as with respect to time (and sometimes the number of servers is also scaled). Our notions of scaling do not involve scaling with respect to time. The definitions of fluid scaling and the HW scaling used in this work are different from that suggested in literature. These scalings are defined below:

Definition 1. *For a sequence of jobs shops under fluid scaling, in job shop* $\mathbf{J}_n, n \in \mathbb{N}$*, the initial number of jobs is* $\{a_n^\omega(j,k) = na^\omega(j,k) : (j,k) \in \boldsymbol{H}, \omega \in \Omega\}$.

Definition 2. *For a sequence of jobs shops under HW scaling, in job shop* $\mathbf{J}_n, n \in \mathbb{N}$*, the initial number of jobs is* $\{a_n^\omega(j,k) = n^2 a^\omega(j,k) : (j,k) \in \boldsymbol{H}, \omega \in \Omega\}$ *and the number of servers is* $\{M_n(i) = nM(i) : i \in \Psi\}$*.*

Under a specific scaling, T_n^ω is the expected makespan of the scaled job shop $\mathbf{J}_n$. Similar notation is used for other quantities. Irrespective of the scaling, from (5.13b) we have the following expression

$$\mathbf{T}_n^\omega = n\mathbf{T}_1^\omega \quad \text{for } n \in \mathbb{N}, \omega \in \Omega, \tag{5.14}$$

where $\mathbf{T}_1^\omega = \mathbf{T}^\omega$ corresponds to the optimal makespan of the unscaled fluid model. For the purpose of analyzing the performance of a policy or heuristic in asymptotic sense, we look at the behavior of expected makespan under fluid and HW scalings. We say a policy π is asymptotically optimal under a specific scaling if:

$$\lim_{n\to\infty} \frac{T_n(\pi)}{n\mathbf{T}} = 1 \tag{5.15}$$

Recall that $\mathbf{T}$ is the optimal makespan of the fluid model. Note that this definition of asymptotic optimality is applicable in both cases, when processing times are non-random or doubly stochastic.

Chapter 6

Single Station Models

We model a single station job shop with each job type being processed exactly once at the station, i.e., there is no reentry. The multiplicity of jobs of each job type is the unique feature compared to the generic single station job shop in scheduling literature. This single job shop model is one of the simplest cases possible. We analyze this base case in this chapter to provide insight for further analysis and illustrate the nature of the policies to be suggested throughout this work.

The station can be comprised of multiple servers. Two different circumstances for the controller are considered. The particular circumstance governs the nature of the scheduling policies. One case explored is when the scheduling problem is an assignment problem. A policy consists of assigning a fixed number of servers to each job class. A simple example of such a policy is assigning a fixed number of resources among different class of customers, in the absence of explicit knowledge regarding the demand for each customer type. However, she is assumed to have a general idea about the distribution of demand. Once a resource is assigned to a particular customer type it cannot be assigned to another customer type due to the huge cost involved in such a transition or fixed costs.

The second case we analyze is a single station problem operating under the class of cyclic policies, $\mathcal{F}_f$. In both cases we formulate the problem and suggest heuristics based on the appropriate fluid models, which turn out to be the same. These heuristics are shown to be asymptotically optimal under the right scaling. When the processing times are doubly stochastic, we also provide probabilistic results for the makespan.

The rest of the chapter is organized as follows, first we provide the formulation for the assignment problem. In subsection 6.1.1, we derive the continuous relaxation of the problem based on which we suggest assignment heuristics and analyze them. In section 6.2, we extend the analysis to doubly stochastic processing times. Next we introduce and analyze the makespan minimization problem under cyclic policies in section 6.3, and suggest a cyclic policy constructed upon optimal solution to the fluid model. In subsection 6.3.1, we study asymptotic properties of the suggested heuristic when the processing times are doubly stochastic.

6.1 Assignment Problem

In this problem each server is assigned to only one job class. We assume the processing times are non-random. Hence, the problem is a type of machine assignment problem. Though an inherent assumption is that the number of servers M is at least equal to the number of job types J, without any loss of generality we assume that $M > J$. The number of servers is assigned before the scenario is realized and not changed. Hence, these policies form a subset of F_f, i.e., the set of constraints CYC are satisfied. Note that no advantage is gained if an allocated server for a particular class is idle, in spite of a job being present. Thus, we restrict the policy set to non-idling policies.

An simplified formulation for the single station assignment problem is provided below. In accordance with the generic model description, the single station problem simplifies to (5.3a)–(5.3f), part of GJS formulation, (i) where $i = 1$ and (ii) the index k is absent, since there is no reentry. An additional constraint that $x(j, l, t)$ is time-independent, restricts the assignment of a server to only a single job class. In this revised formulation $y(j)$ is the number of servers assigned to job class j. The objective function and constraint (5.3b) are retained. Equation (5.3g) becomes redundant as each job belongs to only one particular job class and hence is processed only once. Thus for non-idling policies the formulation for single station assignment model (**SSA**) becomes:

$$\min_{\mathbf{y}} \{\mathbf{E}_\Omega [T^\omega]\}$$

$$\text{s.t.} \quad T^\omega \geq t^\omega(j) \tag{6.1a}$$

$$\sum_{j \in \Phi} y(j) \leq M \tag{6.1b}$$

$$m^\omega(j) X^\omega(j, t) \leq y(j) t \tag{6.1c}$$

$$X^\omega(j, t^\omega(j)) \geq a^\omega(j) \tag{6.1d}$$

$$y(j) \in \mathbb{N}$$

$$j \in \Phi, \omega \in \Omega.$$

Now, combining relations (6.1c) and (6.1d), we have the following equation,

$$t^{\omega}(j) = \left\lceil \frac{a^{\omega}(j)}{y(j)} \right\rceil m(j). \tag{6.2}$$

Note that due to non-idling nature of the policies the difference in the number of jobs processed at two servers is at most one. This observation is an inherent assumption in the equation (6.2). A cycle at a server can be seen as consisting of processing one job of the corresponding job class. Further, there is no idle time in such a cycle. The above optimization problem SSA is a stochastic non-linear program with integer constraints. This in itself might be tough to solve. Hence, a relaxation is suggested in the next subsection. Based on the optimal solution of the relaxation, near optimal heuristic are suggested for the single station assignment problem.

6.1.1 Fluid model

Here we introduce a continuous optimization problem, which is a relaxation of the assignment problem for a single station. When the integer constraints in the above problem, SSA, are relaxed, the resulting optimization problem is a fluid model. Relaxing the integer constraints allows continuous division of the station among all the job classes. All the servers are collapsed into a single server. Simply put, the fluid model is similar to a problem of emptying the tanks of different fluids through a single faucet in the least possible time.

The single station fluid model formulation (**SSF**) is

$$\min_{\mathbf{u}} \left\{ \mathbf{E}_{\Omega} \left[\tilde{T}^{\omega} \right] \right\} \tag{6.3a}$$

$$\text{s.t.} \quad \tilde{T}^{\omega} \geq \tilde{t}(j,\omega)$$

$$\tilde{t}(j,\omega) = \frac{a^{\omega}(j)}{Mu(j)} m(j) \tag{6.3b}$$

$$\sum_{j \in \Phi} u(j) \leq 1 \tag{6.3c}$$

$$u(j) > 0$$

$$j \in \Phi, \omega \in \Omega,$$

where $Mu(j)$ is the fraction of the station that is assigned to process the fluid type j. Note that $\mathbf{T}$ the optimal solution to SSF, is a lower bound on the expected makespan of SSA. In fact, unlike in case of (generic) cyclic policies SSF is an relaxation of SSA. Due to the simple structure of the server assignment model SSA and the associated fluid allocation model SSF, it is reasonable to expect that a *fluid following* allocation policy, i.e., a policy which tries to emulate an optimal solution to the fluid model as closely as possible, would be asymptotically optimal.

Definition 3. *The server assignments $\{y^*(j)\}$ in a fluid following policy $\tilde{\pi}^*$ as a function of job shop parameters are*

$$y^*(j) = \lfloor (M-J)\boldsymbol{u}(j) \rfloor + 1 \quad \textit{for } j \in \Phi \tag{6.4}$$

In the above description, recall that $\{\mathbf{u}(j)\}$ is an optimal solution of SSF. Policy $\tilde{\pi}^*$ does not assume any prior knowledge regarding the realized number of initial jobs of each class. Also policy $\tilde{\pi}^*$ satisfies the constraint (6.1b). The information used in the above allocation of servers is the optimal solution from the fluid model and the scenario independent data $\{M, J\}$. Note that as a server is assigned to only a single job class, this policy $\tilde{\pi}^* \in \digamma_f$.

Having suggested a heuristic, the next stage is to evaluate the performance of the heuristic. In the following result, we derive bounds on the expected makespan, when policy $\tilde{\pi}^*$ is used. Based on these bounds, we establish the asymptotic optimality of $\tilde{\pi}^*$ under HW scaling.

Theorem 8. *The following bounds hold when the allocation policy is* $\tilde{\pi}^*$

$$\boldsymbol{T}\left(\frac{M\dot{\mathbf{u}}}{M\dot{\mathbf{u}} - J + 1}\right) \leq T(\tilde{\pi}^*) \leq \boldsymbol{T}\left(1 + \frac{J}{M-J}\right) + \dot{m} \tag{6.5}$$

where $\dot{\mathbf{u}} := \max_{j\in\Phi}\{\boldsymbol{u}(j)\}$ *and* $\dot{m} := \max_{j\in\Phi}\{m(j)\}$.

Proof. From equation (6.2) for each $\omega \in \Omega$ and $j \in \Phi$ we have,

$$\begin{aligned}
\left\lceil \frac{a^\omega(j)}{(M-J)\mathbf{u}(j)} \right\rceil m(j) &\geq t^\omega(j) \geq \left\lceil \frac{a^\omega(j)}{(M-J)\mathbf{u}(j)+1} \right\rceil m(j) \\
\Rightarrow \left\{ \frac{a^\omega(j)}{(M-J)\mathbf{u}(j)(j)} + 1 \right\} m(j) &\geq t^\omega(j) \geq \frac{a^\omega(j)}{(M-J)\mathbf{u}(j)+1} m(j) \\
\Rightarrow \mathbf{t}^\omega(j)\left(1 + \frac{J}{M-J}\right) + m(j) &\geq t^\omega(j) \geq \mathbf{t}^\omega(j)\left(\frac{M}{M-J+\frac{1}{\mathbf{u}(j)}}\right),
\end{aligned} \tag{6.6}$$

where $\mathbf{t}^\omega(j)$ is the drain time for fluid type j in SSF under optimal allocation $\mathbf{u}$, under scenario ω. The last set of inequalities is implied by relation (6.3b).

For the suggested policy, from (5.4) we have

$$\begin{aligned} T(\tilde{\pi}^*) &= \mathbf{E}_\Omega \left[T^\omega\right] \qquad (6.7)\\ &= \mathbf{E}_\Omega \left[\max_{j\in\Phi} \{t^\omega(j)\}\right]. \end{aligned}$$

Taking the maximum over j and then taking expectations in (6.6) yields the result. □

Theorem 8 implies that in the case of a single station, when the optimal fluid makespan and the number of servers is large along with comparatively smaller number of job classes, the policy $\tilde{\pi}^*$ is near optimal. Next, we look at the asymptotic behavior of the expected makespan under policy $\tilde{\pi}^*$. Under fluid scaling the assignment of servers to different job classes does not change, as a result we might be able to mimic the fluid solution effectively. As the number of servers is scaled in HW scaling we are able to assign servers to closely follow the fluid solution. We readily have the following result.

Theorem 9. *Policy $\tilde{\pi}^*$ is asymptotically optimal for the assignment problem under HW scaling.*

Proof. Under HW scaling the following conditions hold,

$$\mathbf{T}_n \to \infty \text{ as } n \to \infty \qquad (6.8a)$$

$$M_n \to \infty \text{ as } n \to \infty. \qquad (6.8b)$$

Combining Theorem 8, (6.8a) and (6.8b) we see that

$$\lim_{n\to} \frac{T_n(\tilde{\pi}^*)}{\mathbf{T}_n} = 1.$$

Hence, from definition of asymptotic optimality (5.15) the theorem follows. $\square$

Another interesting scaling, linear scaling, is when both the initial number of jobs and number of machines are scaled linearly, i.e., for the single station $M_n = nM$ and $a_n^\omega(j) = na^\omega(j)$. Under this scaling we are able to match the fluid solution closely. For linear scaling from Theorem 8 we have the following result.

Lemma 6. *If policy $\tilde{\pi}^*$ is used for assignment problem in a single station, then under linear scaling*

$$\lim_{n\to\infty} T_n(\tilde{\pi}) \leq \boldsymbol{T} + \dot{m}. \tag{6.9}$$

6.2 Doubly Stochastic Case

Until now the discussion was focused on scenario based entities. However, once the scenario is fixed all the parameters of the job shop are deterministic (though possibly unknown to the controller). Extending the model further, we model random processing times under each scenario. As mentioned in the introduction, the processing times could be random with scenario dependant distributions. We suggest the same policy $\tilde{\pi}^*$ defined in Definition 3.

The randomness of processing times, i.e., the doubly stochastic nature, adds more complexity to the discrete single station model. Even mathematical formulation of the model becomes much more messier. Hence, we look at the continuous relaxation of the single station model as in the previous section. Modifying SSF to include random processing times we get the following formulation (**DSSS**) of the doubly stochastic single station fluid model

$$\min_{\mathbf{u}} \left\{ \mathbf{E}_{\Omega} \left[\mathbf{E}_{\omega} \left[\max_{j \in \Phi} \left\{ \bar{\tilde{t}}^{\omega}(j) \right\} \right] \right] \right\} \tag{6.10a}$$

$$\text{s.t.} \quad \bar{\tilde{t}}^{\omega}(j) = \frac{\sum_{i=1}^{a^{\omega}(j)} \bar{m}_i^{\omega}(j)}{u(j)M} \tag{6.10b}$$

$$\sum_{j \in \Phi} u(j) \leq 1$$

$$u(j) > 0$$

$$j \in \Phi, \omega \in \Omega.$$

Note that for each parameter scenario ω, the makespan for each job class is a random variable. Taking expectations with respect to induced probability measure $\mathbf{P}_{\omega}$ on both sides of (6.10b), reduces to (6.3b), where $\tilde{t}^{\omega}(j) = \mathbf{E}_{\omega}\left(\bar{\tilde{t}}^{\omega}(j)\right)$. Now from Jensen's inequality we have,

$$\mathbf{E}_{\omega} \left(\max_{j \in \Phi} \left(\bar{\tilde{t}}^{\omega}(j) \right) \right) \geq \max_{j \in \Phi} \left(\mathbf{E}_{\omega} \left(\bar{\tilde{t}}^{\omega}(j) \right) \right) = \tilde{T}^{\omega}. \tag{6.11}$$

Hence, in essence it is established that SSF provides a lower bound for DSSS. Also, for the assignment problem DSSS is a relaxation of the discrete single station model with doubly stochastic processing times. So, the optimal solution of SSF is a lower bound on the expected makespan of the single station model. The performance of the suggested heuristic is measured with respect to this lower bound.

Having established a lower bound, we proceed to analyze the performance of the suggested assignment $\tilde{\pi}^*$. Chernoff's bound is used in the following analysis. Lemma below states Chernoff's bound for a set of independent and identically-distributed (iid) random variables.

Lemma 7. *If $\{x_1, x_2, \ldots, x_n\}$ is a set of iid random variables and for some for some $\theta > 0$, $\boldsymbol{E}\left(e^{\theta x_1}\right) < \infty$, then for any $\epsilon > \boldsymbol{E}(x_1)$ and $n \in \mathbb{N}$,*

$$\boldsymbol{P}\left(x_1 + \cdots + x_n \geq n\epsilon\right) \leq e^{-nf(\epsilon)}, \tag{6.12}$$

where $f(\epsilon) := \sup_{\theta}(\theta\epsilon - \log\left(\boldsymbol{E}\left(e^{\theta x_1}\right)\right) > 0$.

Note that $f(\epsilon) > 0$ for any $\epsilon > 0$. The function $f(\cdot)$ is called the Fenchel-Legendre transform of the cumulative generating function and it depends on the distribution of x_1. In the present work, when we consider random processing times to indicate this dependence additional indices like ω and j are used as follows $f^{\omega}(j,\epsilon) = \sup_{\theta}\left(\theta\epsilon - \log\left(\mathbf{E}_{\omega}\left(e^{\theta\bar{m}^{\omega}(j)}\right)\right)\right)$ and $\acute{f}^{\omega}(j,\epsilon) = \sup_{\theta}\left(\theta\epsilon - \log\left(\mathbf{E}_{\omega}\left(e^{-\theta\bar{m}^{\omega}(j)}\right)\right)\right)$.

To aid in analysis, tractability, we construct another policy $\tilde{\pi}^*_{fn}$ from the suggested policy $\tilde{\pi}^*$ as defined below.

Definition 4. *The server assignments in a policy $\tilde{\pi}^*_{fn}$ satisfy $y^*_{fn}(j) = y^*(j)$ for every $j \in \Phi$. Further a maximum of $\tilde{\gamma}^{\omega}(j) := \left\lceil \frac{a^{\omega}(j)}{y^*(j)} \right\rceil$ of class j jobs can be processed at an assigned server.*

Note that this restricted policy $\tilde{\pi}^*_{fn} \in \digamma_{\mathbf{fn}}$. If jobs are always available for processing at a server l, then from Chernoff's bound defining $d := m^{\omega}(j) + \epsilon$, we have,

$$\mathbf{P}_\omega\left(\bar{X}^\omega\left(l,j,\tilde{\gamma}^\omega(j)d\right)>\tilde{\gamma}^\omega(j)\right)=\mathbf{P}_\omega\left(\sum_{i=1}^{\tilde{\gamma}^\omega(j)}\left(\bar{m}_i^\omega(j)\right)\leq\tilde{\gamma}^\omega(j)d\right)$$
$$\geq 1-e^{-\tilde{\gamma}^\omega(j)f^\omega(j,\epsilon)}, \tag{6.13}$$

where $\mathbf{P}_\omega$ is scenario induced probability measure and $f^\omega(j,\epsilon)>0$ for any $\epsilon>0$. Note that the random processing times are properties of jobs and not of the servers and that the restricted policy might introduce forced idleness at an assigned server.

Lemma 8. *When fluid following policy $\tilde{\pi}^*$ is implemented, the makespan under scenario $\omega\in\Omega$, $\bar{T}^\omega$ satisfies the following probabilistic bound for any $\epsilon>0$*

$$\boldsymbol{P}_\omega\left(\bar{T}^\omega\leq\boldsymbol{T}^\omega\frac{M(\ddot{m}^\omega+\epsilon)}{(M-J)\ddot{m}^\omega}+\dot{m}^\omega+\epsilon\right)\geq\left(1-e^{-\ddot{\gamma}^\omega\ddot{f}^\omega(\epsilon)}\right)^{M+J}, \tag{6.14}$$

where $\ddot{m}:=\min_{j\in\Phi}\{m(j)\},\ddot{\tilde{\gamma}}^\omega:=\min_{j\in\Phi}\{\tilde{\gamma}^\omega(j)\}$ *and* $\ddot{f}^\omega(\epsilon):=\min_{j\in\Phi}\{f^\omega(j,\epsilon)\}$.

Proof. From (5.6) we know $T(\tilde{\pi}^*)\leq T(\tilde{\pi}^*_{fn})$. Using this fact and (6.13),

$$\mathbf{P}_\omega\left(\bar{X}^\omega\left(j,\tilde{\gamma}^\omega(j)d\right)=a^\omega(j)\right)\geq\left(\mathbf{P}_\omega\left(\sum_{i=1}^{\tilde{\gamma}^\omega(j)}m_i^\omega(j)\leq\tilde{\gamma}^\omega(j)d\right)\right)^{\lfloor(M-J)\mathbf{u}(j)\rfloor+1}$$
$$\geq\left(1-e^{-\tilde{\gamma}^\omega(j)f^\omega(j,\epsilon)}\right)^{M\mathbf{u}(j)+1}. \tag{6.15}$$

From the definitions of $\tilde{\gamma}^\omega(j)$ and $\tilde{\pi}^*$ we have, for each $\omega \in \Omega$ and each $j \in J$

$$\begin{aligned} \frac{a^\omega(j)}{(M-J)\mathbf{u}(j)+1} \quad &\leq \tilde{\gamma}^\omega(j) \leq \quad \frac{a^\omega(j)}{(M-J)\mathbf{u}(j)}+1 \\ \Rightarrow \frac{\mathbf{t}^\omega(j)}{m^\omega(j)}\frac{M\mathbf{u}(j)}{(M-J)\mathbf{u}(j)+1} \quad &\leq \tilde{\gamma}^\omega(j) \leq \quad \mathbf{t}^\omega(j)\frac{M}{(M-J)m^\omega(j)}+1. \end{aligned} \tag{6.16}$$

Combining (6.15) and (6.16), we obtain,

$$\mathbf{P}_\omega\left(\bar{t}^\omega(j) \leq \mathbf{t}^\omega(j)\frac{M(m^\omega(j)+\epsilon)}{(M-J)m^\omega(j)} + m^\omega(j)+\epsilon\right) \geq \left(1-e^{-\tilde{\gamma}^\omega(j)f^\omega(j,\epsilon)}\right)^{M\mathbf{u}(j)+1}$$

Considering the inequality above across all job classes j, we get (6.14). □

The result above, Lemma 8 states that when policy $\tilde{\pi}^*$ is followed, the probability that the (random) makespan is smaller than the scenario dependant bound is close to one, for job shops in which large number of jobs are being proceeded at a server. Lemma 8 provides an upper bound on makespan under policy $\tilde{\pi}^*$. Next we derive a lower bound.

Lemma 9. *When policy $\tilde{\pi}^*$ is followed, for the makespan the follows bound holds for each $\omega \in \Omega$ and for any $\epsilon > 0$*

$$\boldsymbol{P}_\omega\left(\bar{T}^\omega \geq \boldsymbol{T}^\omega\left(1-\frac{\epsilon}{\ddot{m}^\omega}\right)\right) \geq \left(1-e^{-\ddot{a}^\omega \ddot{f}^\omega(\epsilon)}\right)^J, \tag{6.17}$$

where $\ddot{a}^\omega := \min_{j\in\Phi}\{a^\omega(j)\}$ *and* $\ddot{f}^\omega(\epsilon) := \min_{j\in\Phi}\left\{\acute{f}^\omega(j,\epsilon)\right\}$.

Proof. Since the DSSS is an relaxation of the single station job shop with doubly stochastic processing times, defining $d^{'} := m^\omega(j) - \epsilon$ for each $\omega \in \Omega$

and $j \in \Phi$, we have

$$\begin{aligned}\mathbf{P}_\omega\left(\bar{X}^\omega\left(j, \frac{a^\omega(j)}{y(j)}d^{'}\right) \leq a^\omega(j)\right) &\geq \mathbf{P}_\omega\left(\bar{\bar{t}}^\omega(j) \geq \frac{a^\omega(j)}{\mathbf{u}(j)M}d^{'}\right)\\ &= \mathbf{P}_\omega\left(\sum_{i=1}^{a^\omega(j)} \bar{m}_i^\omega(j) \geq a^\omega(j)d^{'}\right)\\ &= \mathbf{P}_\omega\left(-\sum_{i=1}^{a^\omega(j)} \bar{m}_i^\omega(j) \geq a^\omega(j) - d^{'}\right)\\ &\geq 1 - e^{-a^\omega(j)\acute{f}^\omega(j,\epsilon)}. \end{aligned} \tag{6.18}$$

where $\acute{f}^\omega(j,\epsilon) > 0$ for any $\epsilon > 0$. The last inequality above follows from (6.12). Combining the relations above, (6.16) and (6.18), for each $\omega \in \Omega$ and $j \in \Phi$ we have

$$\mathbf{P}_\omega\left(\bar{t}^\omega(j) \geq \frac{\mathbf{t}^\omega(j)}{m^\omega(j)}\left(m^\omega(j) - \epsilon\right)\right) \geq 1 - e^{-a^\omega(j)\acute{f}^\omega(j,\epsilon)}.$$

Considering the inequality across all job classes j, we get 6.17. □

Now we have upper and lower bounds for makespan under a particular scenario. Combining the bounds (6.14) and (6.17) we immediately get the following theorem.

Theorem 10. *If policy $\tilde{\pi}^*$ is followed, the for each $\epsilon > 0$ and $\omega \in \Omega$*

$$\boldsymbol{P}_\omega\left(lb^\omega \leq \bar{T}^\omega \leq ub^\omega\right) \geq \left(1 - e^{-\ddot{\bar{\gamma}}^\omega \ddot{f}^\omega(\epsilon)}\right)^{M+J} + \left(1 - e^{-\ddot{a}(\omega)\ddot{\acute{f}}^\omega(\epsilon)}\right)^J - 1, \tag{6.19}$$

where

$$lb^\omega := \boldsymbol{T}^\omega\left(1 - \frac{\epsilon}{\ddot{m}^\omega}\right) \quad \text{and} \quad ub^\omega := \boldsymbol{T}^\omega\frac{M(\ddot{m}^\omega + \epsilon)}{(M-J)\ddot{m}^\omega} + \dot{m}^\omega + \epsilon.$$

Next, we show that the policy $\tilde{\pi}^*$ is also asymptotically optimal in terms of the objective function, i.e., total expected makespan. So far we have studied the behavior for each individual sample path. This in itself does not give any picture of the overall convergence of the objective function. To this end, we analyze the convergence properties of the objective function itself. This is necessary to study the behavior of the objective function as the job shop size increases. Before proceeding further, a result regarding convergence of expected value is stated and proved.

Lemma 10. *If $\{x_1, x_2, \ldots\}$ and $\{y_1, y_2, \ldots\}$ are independent sequences of uniformly bounded positive random variables converging to x and y almost surely, then*

(i) $\lim_{n\to\infty} \boldsymbol{E}[\max(x_n, y_n)] = \max(x, y)$.

(ii) $\lim_{n\to\infty} \boldsymbol{E}[\min(x_n, y_n)] = \min(x, y)$.

Proof. Since, the random variables are uniformly bounded and converge w.p.1, from the bounded convergence theorem we can interchange expectation and limit, see [26] for a reference. So, we have

$$\begin{aligned}
\lim_{n\to\infty} \mathbf{E}[\max(x_n, y_n)] &= \mathbf{E}\left[\lim_{n\to\infty} \max(x_n, y_n)\right] \\
&= \mathbf{E}\left[\max\left(\lim_{n\to\infty} x_n, \lim_{n\to\infty} y_n\right)\right] \\
&= \max(x, y).
\end{aligned}$$

The penultimate equality holds because of the continuous mapping theorem. The argument for part (ii) is exactly analogous. $\square$

We now specifically consider HW scaling. We consider the restricted policy $\tilde{\pi}^*_{fn}$ given in Definition 4. Recall that the maximum number of jobs that can be done at an assigned sever is $\tilde{\gamma}^\omega(j)$. Under the policy $\tilde{\pi}^*_{fn}$, let $\bar{m}^\omega_i(j,l)$ be the processing time of i^{th} job processed at assigned server l, and $\bar{\gamma}^\omega(j,l)$ the number of jobs done at the server. Then the makespan satisfies the following inequality,

$$\bar{T}^\omega \leq \max_{\substack{l \in V(j) \\ j \in \Phi}} \left\{ \sum_{i=1}^{\bar{\gamma}^\omega(j,l)} \bar{m}^\omega_i(j,l) \right\}, \tag{6.20}$$

where $V(j)$ denotes the set of servers assigned to job class j. Below $\tilde{\gamma}^\omega_n(j)$ is the maximum number of class j jobs that can be processed at an assigned server in the scaled job shop $\mathbf{J}_n$.

Theorem 11. *If policy $\tilde{\pi}^*_{fn}$ is followed, then under HW scaling*

$$\lim_{n\to\infty} \frac{T_n(\tilde{\pi}^*_{fn})}{n\boldsymbol{T}} = 1.$$

Proof. Due to restriction in the policy $\tilde{\pi}^*_{fn}$ on number of jobs processed, $\bar{\gamma}^\omega_n(j,l) \leq \tilde{\gamma}^\omega_n(j)$, for all $\omega \in \Omega, j \in \Phi$. Using this fact and (6.20), we obtain

$$\lim_{n\to\infty} \mathbf{E}_\omega \left[\frac{\bar{T}^\omega_n}{n} \right] \leq \lim_{n\to\infty} \left(\mathbf{E}_\omega \left[\max_{j\in\Phi} \left\{ \sum_{i=1}^{\tilde{\gamma}^\omega_n(j)} \bar{m}^\omega_i(j) \right\} \left(\frac{1}{n} \right) \right] \right).$$

So, applying the strong law of large numbers and Lemma 10, we have

$$\begin{aligned} \lim_{n\to\infty} \mathbf{E}_\omega \left[\frac{\bar{T}^\omega_n}{n} \right] &\leq \max_{j\in\Phi} \left\{ m^\omega(j) \left(\lim_{n\to\infty} \frac{\tilde{\gamma}^\omega_n(j)}{n} \right) \right\} \\ &= \max_{j\in\Phi} \left\{ \frac{m^\omega(j) a^\omega(j)}{\mathbf{u}(j) M} \right\}. \end{aligned} \tag{6.21}$$

The final equality holds because of HW scaling and relation (6.16). Considering (6.21) over all scenarios, we have

$$\lim_{n\to\infty} \mathbf{E}_\Omega \left[\mathbf{E}_\omega \left[\max_{j\in\Phi} \left\{ \frac{\bar{t}_n^\omega(j)}{n} \right\} \right] \right] \le \mathbf{E}_\Omega \left[\mathbf{T}^\omega \right]. \tag{6.22}$$

(6.22) along with the observation that $\mathbf{T}$ is a lower bound for the objective function yields the result. $\square$

Since $T(\tilde{\pi}^*) \le T(\tilde{\pi}^*_{fn})$, Theorem 11 establishes asymptotic optimality of $\tilde{\pi}^*$ under doubly stochastic processing times.

Under HW scaling we have established:

1. The scaled objective value converges to a lower bound.

2. The makespan under each scenario lines within tight bound of the corresponding optimal fluid makespan with high probability.

6.3 Cyclic Policy

In this section we consider the set of cyclic policies for a single station job shop. First we consider the case when processing times are non-random and then extend the analysis to the case when processing times are doubly stochastic. As in the case of the assignment problem, we suggest a policy, which is based on a optimal solution to the fluid model. The performance of this policy is studied and the policy is shown to be asymptotically optimal under appropriate scaling.

The fluid model FS is same as the model defined by SSF, i.e., the fluid model for both circumstances modeled is the same. One can interpret this result as absence of relevance of circumstance in the context the fluid model, in which, at each station capacity is allocated among the job classes.

We suggest a policy which tries to imitate the optimal solution of the fluid model as closely as possible. We also provide a provision to modify the policy for the case when preemption in not allowed (npr). The suggested policy π_f^* is defined below in terms of parameters $(\mathbf{z}_f, \mathbf{C}_f)$. Note that in this definition of the policy instead of explicitly specifying cycle length we just impose a constraint. Recall from assumption **a4**, $\mathbf{U}$ is an upper bound on processing times.

Definition 5. *The parameters of policy π_f^*, for $j \in \Phi$ and $1 \leq l \leq M$ are,*

$$C_f(l) = C_f \tag{6.23a}$$

$$I_f(j,l) = I_f = \begin{cases} \boldsymbol{U} & \text{if } npr, \\ 0 & o.w. \end{cases} \tag{6.23b}$$

$$z_f(j,l) = z_f(j) = \boldsymbol{u}(j)(C_f - I_f J) \geq \max_{\omega \in \Omega} \{m^{\omega}(j)\}, \tag{6.23c}$$

where $C_f > 0$ is an arbitrary cycle length, $C_f(l)$ is cycle length at server l, and $I_f(j,l)$ and $z_f(j,l)$ are idle time and processing time in a cycle allocated to class j at server l.

For the purpose of *npr*, at the completion of allocated time, a job currently in processing uses the additional idle time I_f for completing processing. In this policy, for a particular job class, we allocate the same fraction of capacity at each server. Also, the cycles at all servers are selected to be of the same reasonable length, as specified by the bound in (6.23c), allowing processing of at least one job of any job class. The below theorem provides bounds on the performance of π_f^*.

Theorem 12. *If policy π_f^* is implemented then,*

$$1 \leq \frac{T(\pi_f^*)}{\boldsymbol{T}} \leq \left(1 + \frac{1}{rC_f - 1}\right) + \frac{C_f}{\boldsymbol{T}}, \tag{6.24}$$

where $r := \min\limits_{\substack{j\in\Phi \\ \omega\in\Omega}} \left\{\frac{\boldsymbol{u}(j)}{m^\omega(j)}\right\}$.

Proof. In scenario $\omega \in \Omega$, the maximum number of class $j \in \Phi$ job that can be completed in a cycle, say $\hat{\gamma}^\omega(j)$, is computed as follows

$$\begin{aligned}
\hat{\gamma}^\omega(j) &= M \left\lfloor \frac{z_f(j,l)}{m^\omega(j)} \right\rfloor \\
&= M \left\lfloor \frac{\mathbf{u}(j)C_f}{m^\omega(j)} \right\rfloor && \text{(6.25a)} \\
\Rightarrow \frac{\mathbf{u}(j)C_f}{m^\omega(j)} - 1 \leq \frac{\hat{\gamma}^\omega(j)}{M} &\leq \frac{\mathbf{u}(j)C_f}{m^\omega(j)}. && \text{(6.25b)}
\end{aligned}$$

However, $\hat{\gamma}^\omega(j)$ jobs can be completed only if there are at least that many jobs to start with. Using the above computation, we get

$$\begin{aligned}
\frac{a^\omega(j)}{\gamma^\omega(j)}C_f \leq \quad & t^\omega(j) && \leq \frac{a^\omega(j)}{\gamma^\omega(j)}C_f + C_f \\
\Rightarrow \frac{a^\omega(j)m^\omega(j)}{\mathbf{u}(j)M} \leq \quad & t^\omega(j) && \leq \frac{a^\omega(j)m^\omega(j)C_f}{M\left(\mathbf{u}(j)C_f - m^\omega(j)\right)} + C_f \\
\Rightarrow \max_{j\in\Phi}\left[\frac{a^\omega(j)m^\omega(j)}{\mathbf{u}(j)M}\right] \leq \quad & T^\omega && \leq \max_{j\in\Phi}\left[\frac{a^\omega(j)m^\omega(j)C_f}{M\left(\mathbf{u}(j)C_f - m^\omega(j)\right)M}\right] + C_f \\
\Rightarrow \mathbf{T} \leq \quad & T(\tilde{\pi}_f) && \leq \max_{\substack{\omega\in\Omega \\ j\in\Phi}}\left(\frac{\mathbf{u}(j)C_f}{\mathbf{u}(j)C_f - m^\omega(j)}\right)\mathbf{T} + C_f.
\end{aligned}$$

Hence, the result follows. □

Among all choices of C_f one particularly interesting choice is one which minimizes the upper bound on the performance measure specified in (6.24). This explicit selection of C_f is also part of defining π_f^*, the single station cyclic policy completely. Using simple calculus we arrive at the following result.

Lemma 11. *If policy π_f^* is implemented and $C_f = \sqrt{\frac{\boldsymbol{T}}{r} + \frac{1}{r}}$, then the upper bound in (6.24) is minimized and*

$$T(\tilde{\pi}_f) \leq \boldsymbol{T} + 2\sqrt{\frac{\boldsymbol{T}}{r} + \frac{1}{r}}. \tag{6.27}$$

We now explore the conditions under which the suggested policy π_f^* with cycle length specified above is asymptotically optimal. We consider a sequence of jobs shops $\{\mathbf{J}_1, \mathbf{J}_2, \ldots\}$. For scaled job shop $\mathbf{J}_n$ belonging to this sequence $\mathbf{T}_n$ is the optimal fluid makespan.

Theorem 13. *Policy π_f^* with cycle length $\sqrt{\frac{\boldsymbol{T}_n}{r} + \frac{1}{r}}$ for scaled single station job shop $\boldsymbol{J}_n$, is asymptotically optimal under HW and fluid scalings.*

Proof. For the cycle length for each scaled job shop $\mathbf{J}_n$, Lemma 5.14 holds. Hence, from (5.14) for both HW and fluid scaling, we have

$$\lim_{n\to\infty} \frac{T_n(\tilde{\pi}_f)}{n\mathbf{T}} = 1. \tag{6.28}$$

□

Unlike the case studied in section 6.1 where servers are simply assigned to the job classes, there is flexibility to assign all job classes to each server in the present model. Due to this flexibility the policy $\tilde{\pi}_f^*$ is asymptotically optimal under fluid scaling also. The choice of cycle length in Theorem 13 for scaled job shop $\mathbf{J}_n$ is just one of many possible values. A set of sufficient conditions to be satisfied by cycle length for asymptotic optimality to hold are introduced in below proposition. The proposition follows immediately from Theorem 12.

Lemma 12. *If policy π_f^* is implemented and cycle length C_f^n for scaled job shosp $\boldsymbol{J}_n$ satisfies*

$$g(n) \leq C_f^n \leq f(n), \tag{6.29}$$

where $f(\cdot)$ and $g(\cdot)$ are real valued functions and

$$\lim_{n\to\infty} \frac{g(n)}{f(n)} = 1 \tag{6.30a}$$

$$\lim_{n\to\infty} \frac{1}{g(n)} = 0 \tag{6.30b}$$

$$\lim_{n\to\infty} \frac{f(n)}{n} = 0, \tag{6.30c}$$

then π_f^* is asymptotically optimal under HW and fluid scalings.

6.3.1 Doubly Stochastic Case

We extend the results in the previous subsection to the case when the processing times are doubly stochastic. The suggested policy in this case is π_f^* as introduced in Definition 5, with a modification. If the processing times are not bounded, then under the *npr* constraint, it is not possible to guarantee a finite makespan. However, recall $\mathbf{U}$ is a uniform upper bound on the processing times. So to guarantee a finite makespan we set $I_f = \mathbf{U}$.

Similar to the description in the previous subsection, let $\bar{\gamma}^\omega(j,l)$ denote the number of jobs completed in a cycle at server l. When jobs are always available, from Chernoff's bound and definition (6.25a) of $\hat{\gamma}^\omega(j)$, for each $\omega \in$

$\Omega, j \in \Phi$ and $1 > \epsilon > 0$ we have

$$\mathbf{P}_\omega \left(\bar{\gamma}^\omega(j,l) \geq \left\lfloor \frac{\hat{\gamma}^\omega(j)}{M}(1-\epsilon) \right\rfloor \right) = \mathbf{P}_\omega \left(\sum_{i=1}^{\check{\gamma}^\omega(j,\epsilon)} \bar{m}_i^\omega(j) \leq z(j) \right)$$
$$\geq \left(1 - e^{-\check{\gamma}^\omega(j,\epsilon) f^\omega(j,\check{\epsilon})}\right)^{\check{\gamma}^\omega(j,\epsilon)}, \tag{6.31}$$

where,

$$\check{\gamma}^\omega(j,\epsilon) := \left\lfloor \frac{\hat{\gamma}^\omega(j)}{M}(1-\epsilon) \right\rfloor \quad \text{and} \quad \check{\epsilon} := \frac{\epsilon}{1-\epsilon}.$$

The next step in our analysis is based on section 6.2. We consider a restricted policy constructed from π_f^*. This restricted policy $\tilde{\pi}_{fn}^*$ is defined below in terms of parameters $(\mathbf{z}_{fn}, \mathbf{C}_{fn})$ and additional constraints. We assume that the *npr* constraint is absent.

Definition 6. *In policy π_{fn}^*, at server $1 \leq l \leq M$ cycle lengths are $C_{fn}(l) = C_f$ and the time allocated to class $j \in \Phi$ at server l is $z_{fn}(j,l) = \boldsymbol{u}(j) C_f$. Further,*

a. A maximum of $\frac{\hat{\gamma}^\omega(j)}{M}$ class j jobs are processed on an assigned server.

b. Only jobs present at the beginning of the cycle are processed.

This restricted policy $\pi_{fn}^* \in \digamma_{fn}$ and the resulting makespan is larger than that under policy π_f^*. Using Chernoff's bound as in Lemma 8 we derive a probabilistic upper bound on makespan when policy π_f^* is followed.

Lemma 13. *If policy $\tilde{\pi}_f^*$ is followed and*

$$C_f \frac{\boldsymbol{u}(j)}{m^\omega(j)} \geq 1 + \frac{1}{1-\epsilon} \quad \forall \quad j \in \Phi, \tag{6.32}$$

for some $\omega \in \Omega$ *and* $\epsilon \in (0,1)$*, then*

$$\boldsymbol{P}_\omega\left(\frac{\bar{T}^\omega}{\boldsymbol{T}^\omega} \leq \dot{ub}^\omega\right) \geq \left(1 - e^{-\ddot{\check{\gamma}}^\omega(\epsilon)\ddot{f}^\omega(\check{\epsilon})}\right)^{M\dot{\check{\gamma}}^\omega(\epsilon)J + \sum\limits_{j\in\Phi} a^\omega(j)}, \tag{6.33}$$

where $\dot{ub}^\omega := \max\limits_{j\in\Phi}\left(\frac{C_f \boldsymbol{u}(j)}{(1-\epsilon)C_f\boldsymbol{u}(j) - (2-\epsilon)m^\omega(j)}\right) + \frac{C_f}{\boldsymbol{T}^\omega}$*,* $\ddot{\check{\gamma}}^\omega(\epsilon) := \min\limits_{j\in\Phi} \check{\gamma}^\omega(j,\epsilon)$ *and* $\dot{\check{\gamma}}^\omega(\epsilon) := \max\limits_{j\in\Phi} \check{\gamma}^\omega(j,\epsilon)$*.*

Proof. If jobs are available, from (6.31), we get the following relation for the number of jobs completed in a cycle $\bar{\hat{\gamma}}^\omega(j)$:

$$\begin{aligned}\mathbf{P}_\omega\left(\bar{\hat{\gamma}}^\omega(j) \geq M\check{\gamma}^\omega(j,\epsilon)\right) &\geq \left(\mathbf{P}_\omega\left(\bar{\gamma}^\omega(1,j) \geq \check{\gamma}^\omega(j,\epsilon)\right)\right)^M \\ &\geq \left(1 - e^{-\check{\gamma}^\omega(j,\epsilon)f^\omega(j,\check{\epsilon})}\right)^{\check{\gamma}^\omega(j,\epsilon)M}.\end{aligned} \tag{6.34}$$

Since the restricted policy π^*_{fn} gives a upper bound on the performance of π^*_f, we obtain the following probabilistic relation from (6.34):

$$\begin{aligned}\mathbf{P}_\omega\left(\bar{X}^\omega\left(j, \left\lceil\frac{a^\omega(j)}{\check{\gamma}^\omega(j,\epsilon)M}\right\rceil C_f\right) = a^\omega(j)\right) &\geq \left(\mathbf{P}_\omega\left(\bar{\hat{\gamma}}^\omega(j) \geq \check{\gamma}^\omega(j,\epsilon)M\right)\right)^{\left\lceil\frac{a^\omega(j)}{\check{\gamma}^\omega(j,\epsilon)M}\right\rceil} \\ &\geq \left(1 - e^{-\check{\gamma}^\omega(j,\epsilon)f^\omega(j,\check{\epsilon})}\right)^{a^\omega(j) + \check{\gamma}^\omega(j,\epsilon)M}.\end{aligned} \tag{6.35}$$

Combining the definition of $\check{\gamma}^\omega(j,\epsilon)$ and (6.25b) we get the following bounds,

$$\begin{aligned}\frac{\hat{\gamma}^\omega(j)}{M}(1-\epsilon) - 1 \quad &\leq \check{\gamma}^\omega(j,\epsilon) \leq \quad \frac{\hat{\gamma}^\omega(j)}{M}(1-\epsilon) \\ \Rightarrow \frac{\mathbf{u}(j)C_f}{m^\omega(j)}(1-\epsilon) - (2-\epsilon) \quad &\leq \check{\gamma}^\omega(j,\epsilon) \leq \quad \frac{\mathbf{u}(j)C_f}{m^\omega(j)}(1-\epsilon).\end{aligned} \tag{6.36}$$

On combining (6.35) and (6.36), after some algebra we obtain:

$$\mathbf{P}_\omega\left(\bar{t}^\omega(j) \le \frac{\mathbf{t}^\omega(j)}{(1-\epsilon) - \frac{(2-\epsilon)m^\omega(j)}{C\mathbf{u}(j)}} + C\right) \ge \left(1 - e^{-\tilde{\gamma}^\omega(j,\epsilon) f^\omega(j,\check{\epsilon})}\right)^{a^\omega(j) + \tilde{\gamma}^\omega(j,\epsilon)M}$$
$$\ge \left(1 - e^{-\ddot{\tilde{\gamma}}^\omega(\epsilon)\ddot{f}^\omega(\check{\epsilon})}\right)^{a^\omega(j)+\dot{\tilde{\gamma}}^\omega(\epsilon)M}. \tag{6.37}$$

Considering 6.37 across all job classes $j \in \Phi$ the lemma follows. □

As in section 6.2, having establishes a probabilistic upper bound we now construct a lower bound in similar manner. We see that due to structure of the fluid model, (6.18) and hence (6.17) hold, even under policy π_f^*. Combining both upper and lower bounds we have the following result.

Theorem 14. *If policy π_f^* is followed and C_f satisfies (6.32), then*

$$\boldsymbol{P}_\omega\left(\dot{lb}^\omega \le \frac{\boldsymbol{T}^\omega}{\bar{T}^\omega(j)} \le \dot{ub}^\omega\right) \ge \left(1 - e^{-\ddot{\tilde{\gamma}}^\omega(\epsilon)\ddot{f}^\omega(\check{\epsilon})}\right)^{M\dot{\tilde{\gamma}}^\omega(\epsilon)J + \sum_{j\in\Phi} a^\omega(j)} + \left(1 - e^{-\ddot{a}(\omega)\ddot{\tilde{f}}^\omega(\epsilon)}\right)^J - 1, \tag{6.38}$$

where $\dot{lb}^\omega := \boldsymbol{T}^\omega\left(1 - \frac{\epsilon}{\ddot{m}(\omega)}\right)$.

Next we establish a convergence result in terms of expectation, showing that the expected makespan under π_f^* is asymptotically optimal. For each server l, consider a sequence of iid random variables $\{\grave{m}_1^\omega(j,l), \grave{m}_2^\omega(j,l), \dots\}$ representing processing times for job class j at server l under scenario ω. Another sequence of iid random variables is, $\{\bar{\gamma}_1^\omega(j,l), \bar{\gamma}_2^\omega(j,l), \dots\}$, where $\bar{\gamma}_1^\omega(j,l)$ is the number of jobs completed in a cycle at server l, assuming jobs are always available for processing. Note that for each server l, $\bar{m}_1^\omega(j,l) \overset{D}{=} \bar{m}^\omega(j)$. From the definitions of $\bar{\gamma}_1^{\omega}(j,l)$ and $\bar{\mathfrak{c}}^{\omega}(j)$ and structure of the policy π_f^*, we have:

$$\bar{\gamma}_1^{\omega}(j,l) = \min\left\{\left\lfloor \frac{\hat{\gamma}(j)}{M} \right\rfloor, \min\left\{n : \sum_{i=1}^{n+1} \bar{m}_i^{\omega}(j) > z_f(j)\right\}\right\} \quad 1 \le l \le M \tag{6.39a}$$

$$\bar{\mathfrak{c}}^{\omega}(j) = \min\left\{n : \sum_{1}^{n} \bar{\bar{\gamma}}_i^{\omega}(j) \ge a^{\omega}(j)\right\}, \tag{6.39b}$$

where $\bar{\bar{\gamma}}_1^{\omega}(j) = \sum_{l=1}^{M} \bar{\gamma}_1^{\omega}(j,l)$.

Note that in (6.39a), for server l and job class j, under scenario ω and for some $n \geq \mathbb{N}$, the event $\{\bar{\gamma}_1^{\omega}(j) + 1 = n\}$ is independent of the sequence $\left\{\bar{m}_{n+1}^{\omega}(j), \bar{m}_{n+2}^{\omega}(j), \ldots\right\}$. So, we observe that $\bar{\gamma}_1^{\omega}(j,l) + 1$ is stopping time for the sequence of random processing times. Similarly, $\bar{\mathfrak{c}}^{\omega}(j)$ is stopping time for the sequence of number of jobs completed in a cycle. Using these definitions we derive a lower bound on the expected makespan under policy π_f^*.

Lemma 14. *If policy π_f^* is followed and $\frac{z(j)}{m^{\omega}(j)} > 1$ for all $j \in \Phi, \omega \in \Omega$ then,*

$$\boldsymbol{E}_{\omega}(\bar{t}^{\omega}(j)) \leq \frac{\boldsymbol{t}^{\omega}(j)}{1 - \frac{m^{\omega}(j)}{C_f \boldsymbol{u}(j)}} + C_f, \tag{6.40}$$

for each $\omega \in \Omega$.

Proof. From 6.39a and 6.39b we observe that the stopping times are. Further, $\bar{m}_1^{\omega}(j) \geq 0$, so we can apply *Wald's equation* to both the sequences mentioned here, see [24] for reference. Hence, from (6.39a) under each scenario ω we get:

$$\begin{aligned}
\mathbf{E}_\omega\left[\sum_{i=1}^{\grave{\gamma}_1^\omega(j,l)+1}\bar{m}_i^\omega(j)\right] &= \mathbf{E}_\omega\left[\grave{\gamma}_1^\omega(j)+1\right]\mathbf{E}_\omega\left[\bar{m}_i^\omega(j)\right] \\
\Rightarrow \mathbf{E}_\omega\left[\bar{\gamma}_1^\omega(j,l)\right] &> \frac{z_f(j)}{m^\omega(j)} - 1 \\
\Rightarrow \mathbf{E}_\omega\left[\bar{\bar{\gamma}}_1^\omega(j)\right] &> M\frac{z_f(j)}{m^\omega(j)} - M. \qquad (6.41)
\end{aligned}$$

From (6.39b), applying Wald's equation, for scenario ω we have

$$\begin{aligned}
\mathbf{E}_\omega\left[\bar{\mathfrak{c}}^\omega(j)\right]\mathbf{E}_\omega\left[\bar{\bar{\gamma}}_1^\omega(j)\right] &= \mathbf{E}_\omega\left[\sum_{i=1}\bar{\mathfrak{c}}^\omega(j)\bar{\bar{\gamma}}_i^\omega(j)\right] \\
&= \mathbf{E}_\omega\left[\sum_{i=2}\bar{\mathfrak{c}}^\omega(j)\bar{\bar{\gamma}}_i^\omega(j)\right] + \mathbf{E}_\omega\left[\bar{\bar{\gamma}}_1^\omega(j)\right] \\
&< a^\omega(j) + \mathbf{E}_\omega\left[\bar{\bar{\gamma}}_1^\omega(j)\right] \\
\Rightarrow \mathbf{E}_\omega\left[\bar{\mathfrak{c}}^\omega(j)\right] &< \frac{a^\omega(j)}{M\left(\frac{z_f(j)}{m^\omega(j)}-1\right)} + 1. \qquad (6.42)
\end{aligned}$$

The final inequality above follows from (6.41). From the definition of $z_f(j)$ under policy π_f^* and (6.42) the result follows. □

The policy π_f^* still has an undefined parameter C_f (cycle length). We choose a cycle length sequence appropriately for a sequence of scaled job shops $\{\mathbf{J}_1, \mathbf{J}_2, \ldots\}$ under fluid or HW scaling. Recall C_f^n corresponds to cycle length under policy π_f^* for scaled job shop $\mathbf{J}_n$.

Theorem 15. *If policy π_f^* is followed and the cycle length sequence $\left\{C_f^1, C_f^2, \ldots\right\}$ satisfies*

$$\lim_{n\to\infty}\frac{C_f^n}{n} = 0 \quad \textit{and} \quad \lim_{n\to\infty}\frac{1}{C_f^n} = 0, \qquad (6.43)$$

then under HW and fluid scalings

$$\lim_{n\to\infty} \frac{T_n(\pi_f^*)}{n} = \boldsymbol{T}.$$

Proof. Under HW and fluid scaling $\mathbf{T}_n^\omega = n\mathbf{T}^\omega$. Hence, Lemma 14 and Lemma 10 yield:

$$\begin{aligned}
\limsup_{n\to\infty} \mathbf{E}_\omega\left[\frac{\bar{T}_n^\omega}{n}\right] &= \lim_{n\to\infty} \mathbf{E}_\omega\left[\max_{j\in\Phi}\left\{\frac{\bar{t}_n^\omega}{n}\right\}\right] \\
&= \max_{j\in\Phi}\left\{\mathbf{E}_\omega\left[\lim_{n\to\infty}\frac{\bar{t}_n^\omega}{n}\right]\right\} \\
&\leq \max_{j\in\Phi}\left\{\lim_{n\to\infty}\left(\frac{\mathbf{t}^\omega(j)}{1-\frac{m^\omega(j)}{C_f^n\mathbf{u}(j)}} + \frac{C_f^n}{n}\right)\right\} \\
&= \max_{j\in\Phi}\left\{\mathbf{t}^\omega(j)\right\}. \qquad (6.44)
\end{aligned}$$

The final equality holds because of the condition (6.43). Since all the jobs have to completed, we have the following lower bound of $\bar{T}_n^\omega$

$$\begin{aligned}
\bar{T}_n^\omega &\geq \max_{j\in\Phi}\left\{\sum_{i=1}^{a_n^\omega(j)}\frac{\bar{m}_i^\omega(j)}{u(j)M}\right\} \\
\Rightarrow \liminf_{n\to\infty}\mathbf{E}_\omega\left[\frac{T_n^\omega}{n}\right] &\geq \lim_{n\to\infty}\mathbf{E}_\omega\left[\max_{j\in\Phi}\left\{\sum_{i=1}^{a_n^\omega(j)}\frac{\bar{m}_i^\omega(j)}{u(j)M}\right\}\right] \\
&= \max_{j\in\Phi}\left\{\mathbf{t}^\omega(j)\right\}, \qquad (6.45)
\end{aligned}$$

where the final equality follows from Lemma 10. From (6.44) and (6.45), the result follows. □

Chapter 7

General Job Shop

In the previous chapter we introduced the makespan problem for single stations operating under a cyclic policy. Next, we extensively studied the behavior of fluid following policies and established their asymptotic optimality. Apart from scheduling under cyclic policies, we also examined the problem of assigning machines to job types. We have showed that under appropriate scaling, the suggested heuristic is asymptotically optimal. The next natural progression is to consider the generic job shop introduced in Chapter 5. Each job follows a fixed route through possibly different stations. A station can be comprised of multiple servers.

A job's route may be reentrant, i.e., the job can visit the same station at different stages of its route. One typical example is the route of a specific product in semiconductor wafer fabrication. There is an enormous amount of research showing how the reentrant property of a system makes scheduling problems, and even stability analysis more complex. In [19], the author introduces the term 'reentrant' and provides a nice survey of results concerning scheduling and stability of reentrant lines. In [11], Dai and Weiss address the question of stability or instability of reentrant lines through analysis of limiting fluid models. The fact that stability of even a simple reentrant line, , under the usual traffic conditions, is not guaranteed is shown by Lu and Kumar [20]. In [28], it is shown that even for a simple reentrant line with two machines, the makespan minimization problem is NP-hard. However, we consider problems in which there is a high multiplicity of similar jobs, which makes development of an asymptotically optimal heuristic from the corresponding fluid model possible. Scheduling of such systems with high multiplicity using fluid models, has been of particular interest in the queueing literature. Finding optimal controls for a broader class of fluid models (multi-class) is an active area of research. In [9, 12] the authors deal with scheduling of fluid networks, while fluid networks are used to schedule discrete networks in [4, 5, 6, 10, 21] and [22].

The analysis in this chapter to a great extent borrows tools and arguments from Chapter 6. We construct a restricted version of the proposed cyclic policy and establish its asymptotically optimality. Conditions are derived for which the policy is also asymptotically optimal under specific conditions, even when the processing times are scenario dependent random variables.

The rest of the chapter is organized in the following format. In section 7.1, we suggest a heuristic, prove its asymptotic optimality. In section 7.2 we extend the analysis to the case of random processing times. We look at the probabilistic behavior of makespan under each scenario. In subsection 7.2.1, we show that the fluid following policy is asymptotically optimal. Finally, we discuss the results and further research direction in the final section 7.3.

7.1 Multiple Station Job Shops

First recall that in Chapter 6 we dropped the index k which represents the stage in a job's route. We reintroduce this index in the following analysis. The job shop consists of $\{1, \cdots, I\}$ stations processing $\{1, \cdots, J\}$ jobs types. Each job type j has a fixed route with $K(j)$ steps in which the k^{th} step is done done at station $\rho(j,k)$. Station i can process a set of job classes $\sigma(i)$. As in the previous chapter we assume that certain parameters, the initial number of jobs and processing times, are unknown to the controller. Recall that Ω is the set of all possible outcomes for this set of parameters, and that ω represents a single outcome. Everything is deterministic within a single scenario ω, until the doubly stochastic case is considered.

The heuristic we suggest is π_f^*, an extension of the policy suggested in the single station job shop case and given by (6.23a)–(6.23b). The parameters $(\mathbf{z}_f, \mathbf{C}_f)$ defining the policy are (i) the cycle length $C_f(l,i)$ for each server $l \in \varpi(i)$ of station $i \in \Psi$, (ii) the time allocations in a cycle $z_f(j,k,l)$ and $I_f(j,k,l)$, at each server $l \in \varpi(\rho(j,k))$ for each job class $(j,k) \in \mathbf{H}$. We define the policy below in terms of the aforementioned parameters.

Definition 7. *The parameters of policy π_f^* are*

$$C_f(l,i) = C_f \tag{7.1}$$

$$I_f(j,k,l) = I_f = \begin{cases} \boldsymbol{U} & \textit{if npr}, \\ 0 & \textit{o.w.} \end{cases} \tag{7.2}$$

$$z_f(j,k,l) = z_f(j,k) = \boldsymbol{u}(j,k)(C_f - I_f) \geq \max_{\omega\in\Omega} m^\omega(j,k), \tag{7.3}$$

where $C_f > 0$ is an arbitrary cycle length, $l \in \varpi(i), i \in \Psi$ and $(j,k) \in \boldsymbol{H}$.

Before proceeding further, we introduce additional notation used in the subsequent analysis. $Q^{\omega}_{s}(j,k)$ is the queue length at the beginning of cycle $s \in \mathbb{N}$ for job class $(j,k) \in \mathbf{H}$, and $r^{\omega}_{s}(j,k)$ is the number of class (j,k) jobs completed in cycle s. We define $\mathfrak{c}^{\omega}(j,k)$ to be the number of cycles needed to process all jobs of a particular class. These definitions can be mathematically expressed as follows:

$$Q^{\omega}_{s}(j,k) = a^{\omega}(j,k) - X^{\omega}(j,k,(s-1)C) + X^{\omega}(j,k-1,(s-1)C) \tag{7.4a}$$

$$r^{\omega}_{s}(j,k) = X^{\omega}(j,k,sC) - X^{\omega}(j,k,(s-1)C) \tag{7.4b}$$

$$\omega \in \Omega, (j,k) \in \mathbf{H}, s \in \mathbb{N}, 1 \leq k \leq K(j).$$

We derive a relation between the processes above. Since $X^{\omega}(j,k,0) = 0$ for all $(j,k) \in \mathbf{H}$, recursively expanding the relation (7.4b), we obtain

$$X^{\omega}(j,k,sC) = X^{\omega}(j,k,(k-1)C) - \sum_{d=k}^{s} r^{\omega}_{d}(j,k) \quad \text{for } s \geq k \geq 1. \tag{7.5}$$

We construct a restricted version of the policy π^*_f, say $\pi^*_{fi} \in \digamma_{fi}$. Observe that though the controller is assumed to have additional information for implementation of π^*_{fi}, from (5.6) the makespan under the restricted policy is higher compared to that under π^*_f. The complete definition of π^*_{fi} is provided below.

Definition 8. *The parameters $(\boldsymbol{z}_{fi}, \boldsymbol{C}_{fi})$ of policy π^*_{fi} are such that $\boldsymbol{z}_{fi} = \boldsymbol{z}_f$ and $\boldsymbol{C}_{fi} = \boldsymbol{C}_f$ where the equality is componentwise. Further,*

a. *Processing of jobs in job class (j,k) starts only after cycle $k-1$.*

b. *Only the jobs present in the queue at beginning of a cycle are processed* *during the cycle.*

As in relation (6.25a), the maximum number of jobs of class (j,k) that can be completed in a cycle, $\gamma^\omega(j,k)$ is given by the following:

$$\gamma^\omega(j,k) := M(\rho(j,k)) \left\lfloor \frac{\mathbf{u}(j)C}{m(\omega,j)} \right\rfloor . \tag{7.6}$$

However, it is possible that there are not that many jobs present. So the number of jobs completed in a given cycle $s \geq k+1$ for $(j,k) \in \mathbf{H}$ is

$$\begin{aligned} r_s^\omega(j,k) &= \min\left[Q_s(\omega,j,k), \hat{\gamma}^\omega(j,k)\right] \\ \Rightarrow r_s^\omega(j,k) &= \min\left[a^\omega(j,k) - \sum_{d=k}^{s-1} r_d^\omega(j,k) + \sum_{d=k-1}^{s-1} r_d^\omega(j,k-1), \hat{\gamma}^\omega(j,k)\right]. \end{aligned} \tag{7.7}$$

The second equality uses equation (7.5). Based on (7.7), we arrive at the following lemma. This lemma essentially states that if the number of jobs being processed at the immediate upstream station is non-increasing, then so is the number of jobs processed at the given station. The result is intuitive. Rephrased, Lemma 15 establishes that once the queue size of a job class falls below a threshold it does not increase. This is helpful in establishing the asymptotic results.

Lemma 15. *Under policy π^*_{fi}, if for job type $j \in \Phi$, $r_s^\omega(j,k-1)$ is non-increasing in s for $s > k-1$ and $2 \leq k \leq K(j)-1$, then, $r_s^\omega(j,k)$ is non-increasing in s.*

Proof. If $s^{'} \geq k$ is the first cycle when the number of jobs processed in the cycle is less than $\grave{\gamma}^\omega(j,k)$, then $\grave{\gamma}^\omega(j,k) > r_{s^{'}}^\omega(j,k) \geq r_{s^{'}-1}^\omega(j,k-1)$, otherwise the number of jobs in queue at the beginning of cycle $s^{'}$ is greater than $\grave{\gamma}^\omega(j,k)$, which contradicts the assumption. Hence, as $r_s^\omega(j,k-1)$ is non-increasing in s for $s \geq k-1$, we have

$$\begin{aligned} &r_s^\omega(j,k-1) < \grave{\gamma}^\omega(j,k) \quad \text{ for } s \geq s^{'} - 1 \\ &\Rightarrow r_{s+1}^\omega(j,k) = r_s^\omega(j,k-1) \quad \text{ for } s \geq s^{'} - 1, \end{aligned}$$

were equality follows from (7.7). So, $r_s^\omega(j,k)$ is non-increasing in s $(s \geq k)$ as $r_s^\omega(j,k-1)$ is non-increasing in s $(s \geq k-1)$. □

Jobs of type j, cannot complete step k, until they have finished processing at the previous station. Further, jobs being completed at a station cannot be processed at the next station in the present cycle. Hence we have, $\mathfrak{c}^\omega(j,k) \geq \mathfrak{c}^\omega(j,k-1)+1$. We use this observation in the next lemma. We show that, if the dynamics of a job shop are such that the number of jobs of class (j,k) processed in a cycle $s \leq \mathfrak{c}^\omega(j,k-1)$, falls below a threshold, then the class empties immediately after its predecessor class empties.

Lemma 16. *Under policy π_{fi}^*, if $r_s^\omega(j,k) < \gamma^\omega(j,k)$ and $\mathfrak{c}^\omega(j,k-1) \geq s$ for some $k \leq s \leq K(j)$ and $j \in \Phi$, then $\mathfrak{c}^\omega(j,k) = \mathfrak{c}^\omega(j,k-1)+1$.*

Proof. First note that $c^\omega(j,k) \geq c^\omega(j,k-1)+1$. Further, due to Lemma 15 and the fact that $r_s^\omega(j,k) < \grave{\gamma}^\omega(j,k)$, we obtain

$$\begin{aligned} &r_{\mathfrak{c}^\omega(j,k-1)+1} < \gamma^\omega(j,k) \\ \Rightarrow\ &Q_{\mathfrak{c}^\omega(j,k-1)+2} = r_d^\omega(j,k-1), \quad d = \mathfrak{c}^\omega(j,k-1)+1 \\ \Rightarrow\ &Q_{\mathfrak{c}^\omega(j,k-1)+2} = 0 \\ &\Rightarrow \mathfrak{c}^\omega(j,k) = \mathfrak{c}^\omega(j,k-1)+1. \end{aligned} \tag{7.8}$$

The second implication follows from the fact that after $\mathfrak{c}^\omega(j,k-1)$ cycles there are no jobs of class $(j,k-1)$ left to be processed. $\square$

When there is no upstream station, the expression for the number of cycles required is

$$\mathfrak{c}^\omega(j,1) = \left\lceil \frac{a^\omega(j,1)}{\hat{\gamma}^\omega(j,2)} \right\rceil . \tag{7.9}$$

This is a special case ($k = 1$) of the following lemma. The lemma pertains to the situation when the job shop is such that, for class (j,k) the maximum number of jobs possible are processed in each cycle up till $\mathfrak{c}^\omega(j,k-1)$. In such circumstances we show that the number of cycles needed to deplete all the jobs in class (j,k) is equal to number of cycles needed when the total initial workload is available to the station without any delay.

Lemma 17. *When policy π^*_{fi} is implemented, if for job class $(j,k) \in \mathbf{H}, \mathbf{r}^\omega_\mathbf{d}(\mathbf{j},\mathbf{k}) = \grave{\gamma}^\omega(\mathbf{j},\mathbf{k})$ where $d = \mathfrak{c}^\omega(j,k-1)+1$, then*

$$\mathfrak{c}^\omega(j,k) = \left\lceil \frac{b^\omega(j,k)}{\grave{\gamma}^\omega(j,k)} \right\rceil + k - 1. \tag{7.10}$$

Proof. Firstly, note that if $\mathfrak{c}^\omega(j,k-1)+1 \le s \le \mathfrak{c}^\omega(j,k)$ then,

$$\begin{aligned} r^\omega_s(j,k) &< \grave{\gamma}^\omega(j,k) \\ \Rightarrow Q^\omega_{s+1}(j,k) &= 0 \\ \Rightarrow \mathfrak{c}^\omega(j,k) &= s. \end{aligned}$$

So, we have the following equation

$$r^\omega_s(j,k) = \check{\gamma}^\omega(j,k) \quad \text{for } k \le s < \mathfrak{c}^\omega(j,k). \tag{7.11}$$

Also, for the final cycle since all jobs are completed, we have $r_s^\omega(j,k) \le \check{\gamma}^\omega(j,k)$. Define $d^{'} := c^\omega(j,k)$. From definition of $c^\omega(j,k)$ and (7.7), we get

$$r_{d'}^\omega(j,k) = a^\omega(j,k) - \sum_{d=k}^{d^{'}-1} r_d^\omega(j,k) + b^\omega(j,k-1).$$

Combining this equation with (7.11), we obtain

$$\begin{aligned} r_{d'}^\omega(j,k) &= b^\omega(j,k) - \left(d^{'} - k + 1\right)\grave{\gamma}^\omega(j,k) \\ \Rightarrow \grave{\gamma}^\omega(j,k)\left(d^{'} - k + 2\right) &\ge b^\omega(j,k) \ge \left(d^{'} - k + 1\right)\grave{\gamma}^\omega(j,k). \end{aligned}$$

The result follows from the above inequality. $\square$

The two lemmas provide us with expressions for the number of cycles needed under two different situations that may unfold. In the next lemma, we establish that the maximum of these two expression is actually the number of cycles needed to complete processing of all jobs at class (j,k). Note that $\mathfrak{c}^\omega(j,k) = 0$ if $b^\omega(j,k) = 0$.

Lemma 18. *The number of cycles needed to complete all the jobs requiring processing at class $(j,k) \in \mathbf{H}$, i.e., $b^\omega(j,k) > 0$ jobs, under policy π_{fi}^* is*

$$\mathfrak{c}^\omega(j,k) = \max\left\{\mathfrak{c}^\omega(j,k-1) + 1, \left\lceil \frac{b^\omega(j,k)}{\grave{\gamma}^\omega(j,k)} \right\rceil + k - 1\right\}, \tag{7.13}$$

where $c^\omega(j,0) = 0$.

Proof. From Lemmas 16 and 17, we know that $\mathfrak{c}^\omega(j,k)$ is equal to one of the expression over which the maximum is considered. Further, $\mathfrak{c}^\omega(j,k-1) + 1$ is a lower bound on $\mathfrak{c}^\omega(j,k)$. Hence, if $\mathfrak{c}^\omega(j,k-1) + 1 \ge \left\lceil \frac{b^\omega(j,k)}{\grave{\gamma}^\omega(j,k)} \right\rceil + k - 1$, the result (7.13) follows.

Recall that under policy π^*_{fi} jobs of class (j,k) cannot be processed before cycle k. Hence, assuming all the jobs of class (j,k) are available without any delay, the total number of cycles needed to complete $b^\omega(j,k)$ jobs is given by (7.10). However, since jobs are to be processed at upstream station first, not all jobs might be available when required, i.e., $c^\omega(j,k) \geq \left\lceil \frac{b^\omega(j,k)}{\dot{\gamma}^\omega(j,k)} \right\rceil + k - 1$. Now having shown that the second term is also a lower bound, arguing as above it is easy to derive the result (7.13). $\square$

Having established an explicit expression for the number of cycles needed to complete processing of a job class, we can derive an expression for the total number of cycles required to process all jobs. Recall $\mathbf{u}(j,k)$ is the allocation to class $(j,k) \in \mathbf{H}$ in an optimal solution. We use the expression for $\mathfrak{c}^\omega(j,K(j))$ to prove the asymptotic optimality of π^*_{fi}. Asymptotic optimality of the suggested policy π^*_f follows.

Theorem 16. *When the fluid following policy π^*_f is implemented, the following bounds hold:*

$$\boldsymbol{T} \leq T(\pi^*_f) \leq \boldsymbol{T}\left(1 + \frac{1}{rC_f - 1}\right) + \dot{K}C_f, \tag{7.14}$$

where $r = \min\limits_{\substack{\omega\in\Omega \\ (j,k)\in\boldsymbol{H}}} \left\{\frac{\boldsymbol{u}(j,k)}{m^\omega(j,k)}\right\}$ and $\dot{K} = \max\limits_{j\in\Phi}\{K(j)\}$.

Proof. Considering Lemma 7.10 across the complete route of a particular job type j and recursively applying (7.13), we obtain

$$\begin{aligned}
\mathfrak{c}^\omega(j, K(j)) &= \max\left\{\mathfrak{c}^\omega(j, K(j)-1)+1, \left\lceil\frac{b^\omega(j,K(j))}{\gamma^\omega(j,K(j))}\right\rceil + K(j)-1\right\} \\
&= \max_{1\le p\le K(j)}\left\{\left\lceil\frac{b^\omega(j,p)}{\gamma^\omega(j,p)}\right\rceil + K(j)-1\right\} \\
&\le \max_{1\le p\le K(j)}\left\{\frac{b^\omega(j,p)}{\gamma^\omega(j,p)}\right\} + K(j) \\
&\le \max_{1\le p\le K(j)}\left\{\frac{b^\omega(j,p)}{\frac{\mathbf{u}(j,k)M(\rho(j,k))C_f}{m^\omega(j,k)}-1}\right\} + K(j) \\
&\le \left(1+\frac{1}{rC_f-1}\right)\max_{p\in\Upsilon(j)}\left\{\frac{b^\omega(j,p)m^\omega(j,k)}{\mathbf{u}(j,k)M(\rho(j,k))C}\right\} + \dot{K} \\
&\le \frac{\mathbf{t}\omega(j,k)}{C}\left(1+\frac{1}{rC-1}\right) + \dot{K}. \qquad (7.15)
\end{aligned}$$

The above derivation of inequalities explicitly uses the definitions of r and $\dot{\gamma}(\cdot)$. The makespan for a particular job type is $\mathfrak{c}^\omega(j, K(j))C$. From (7.15) and the fact that $\mathbf{T}$ is a lower bound, the result (7.14) follows. $\square$

Note that C_f is a free parameter in the policy π_f^* with a restriction (7.3). As in section 6.3, we give the specific choice C_f which minimizes the upper bound in (7.14). The result given below can be derived using simple calculus.

Lemma 19. *If policy π_f^* is implemented and cycle length is chosen such that $C_f = \sqrt{\frac{\boldsymbol{T}}{r\dot{K}}} + \frac{1}{r}$ then upper bound in (7.14) is minimized and*

$$T(\pi_f^*) \le \boldsymbol{T} + 2\sqrt{\frac{\dot{K}\boldsymbol{T}}{r}} + \frac{\dot{K}}{r}. \qquad (7.16)$$

Next we analyze the behavior of the policy π_f^* with cycle length suggested above, under HW and fluid scaling. The theorem below can be derived in the same way as Theorem 13 and so is stated without a proof.

Theorem 17. *Under HW and fluid scaling the policy π_f^* with the cycle length $\sqrt{\frac{\boldsymbol{T}_n}{r}} + \frac{1}{r}$ for the scaled general job shop $\boldsymbol{J}_n$, is asymptotically optimal.*

We have proven asymptotic optimal of the policy π_f^* when a specific value of C_f is chosen. However, in general as long as C_f satisfies certain conditions, policy π_f^* is asymptotically optimal. Recall that C_f^n corresponds to the cycle length suggested under policy π_f^* for the scaled general job shop $\mathbf{J}_n$. In the theorem below, we give a set of sufficient conditions under which the policy π_f^* is asymptotically optimal.

Theorem 18. *If policy π_f^* is followed and $\left\{C_f^1, C_f^2, \ldots\right\}$ the cycle length sequence satisfies (6.29)–(6.30c), then under HW and fluid scalings*

$$\lim_{n\to\infty} \frac{T_n(\pi_f^*)}{\boldsymbol{T}} = n. \tag{7.17}$$

7.2 Doubly Stochastic Case

Until now we considered job shop models which were deterministic for a fixed scenario $\omega \in \Omega$. In this subsection we study the asymptotic properties of the policy π_{fi}^* and in turn that of π_f^* when processing times are random for a given scenario ω. When the cycle length satisfies certain conditions, we have shown the policy π_f^* to be asymptotically optimal for a single station job shop with random processing times. Though more complex, the analysis in this section is similar to that in subsection 6.3.1.

Extending the definition given in equation (6.31) to the general job shop setting, for a job class (j,k), $\check{\gamma}^\omega(j,k,\epsilon)$ is defined as

$$\check{\gamma}^\omega(j,k,\epsilon) := \left\lfloor \frac{\hat{\gamma}^\omega(j,k)}{M(\rho(j,k)}(1-\epsilon) \right\rfloor \quad \text{and} \quad \check{\epsilon} := \frac{\epsilon}{1-\epsilon}. \tag{7.18}$$

Other notation is likewise extended in a straightforward manner to the general job shop setting. Assuming jobs are always available, applying Chernoff's bound, (6.12), to the number of jobs of class (j,k) processed in scenario ω, we have for each $\epsilon \in (0,1)$,

$$\mathbf{P}_\omega\left(\bar{\hat{\gamma}}^\omega(j,k) \geq M(\rho(j,k))\check{\gamma}^\omega(j,k,\epsilon)\right) \geq \left(1 - e^{-\check{\gamma}^\omega(j,k,\epsilon) f^\omega(j,k,\check{\epsilon})}\right)^{M(\rho(j,k))}. \tag{7.19}$$

We consider the dynamics of a 'parallel' job shop, say JS2, similar to that in the previous section, but operating under a policy with an additional restriction compared to policy π_f^*. In JS2 the processing times are non-random and equal to the means, i.e., $m^\omega(j,k)$ is the processing time of job class (j,k) under scenario ω. We consider JS2, to make the comparison of the dynamics of the job shop with that of fluid model tractable. The restricted policy π_ϵ^* of JS2 is defined below in terms of parameters $(\mathbf{z}_\epsilon, \mathbf{C}_\epsilon, \epsilon)$, policy π_f^* and some additional restrictions.

Definition 9. *For each $\epsilon \in (0,1)$, the parameters of policy π_ϵ^* are $\boldsymbol{z}_\epsilon = \boldsymbol{z}_f$ and $\boldsymbol{C}_\epsilon = \boldsymbol{C}_f$ where the equality is componentwise. Further,*

a. *Processing of jobs in the job class (j,k) starts only after cycle $k-1$.*

b. *Only the jobs present in the queue at the beginning of a cycle are processed during the cycle.*

c. *In a cycle a maximum of $\check{\gamma}^\omega(j,k,\epsilon)$ of class (j,k) jobs are processed at a* *server.*

The additional restriction (**c**) forces the policy to depend on ϵ. To represent this association an additional index or superscript ϵ is used. For example, while $r_s^\omega(j,k)$ is the number of class (j,k) jobs processed in cycle s under policy π_{fi}^*; when additionally (**c**) is enforced the quantity is represented by $r_s^\omega(j,k,\epsilon)$. A similar interpretation is used for other quantities.

The following relation gives a probabilistic bound on the number of jobs that can be processed during a given cycle. From (7.4b), we have

$$\begin{aligned}&\mathbf{P}_\omega\left(\bar{r}_{s+1}^\omega(j,k) = \min\left\{\bar{Q}_s^\omega(j,k), M(\rho(j,k))\check{\gamma}^\omega(j,k,\epsilon)\right\}\right)\\&\quad\geq \left(1-e^{-\check{\gamma}^\omega(j,k,\epsilon)f^\omega(j,k,\check{\epsilon})}\right)^{M(\rho(j,k)} \geq \left(1-e^{-\ddot{\gamma}^\omega(\epsilon)\ddot{f}(\omega,\check{\epsilon})}\right)^{M(\rho(j,k))}.\end{aligned} \tag{7.20}$$

We prove that for job shop with large number of jobs, under each scenario ω with high probability the makespan under policy π_{fi}^* ia within tight (ϵ-dependent) bounds of the associated optimal fluid makespan. For this purpose we consider conditional probabilities in the ensuing analysis. Until specified again in the analysis we restrict to a fixed job type $j \in \Phi$ and fixed scenario $\omega \in \Omega$. We define certain events for this fixed job type scenario pair (j,ω). For each cycle $s \geq k, K(j) \geq k$:

$$E(k,s) := \left\{\bar{X}^\omega(j,k,sC) \geq X^\epsilon\omega(j,k)\right\} \qquad \text{for } k \geq 1 \tag{7.21}$$

$$\begin{aligned}\mathbf{K}(k,s) &:= \left\{(\hat{k},\hat{s}) : 1 \leq \hat{k} \leq k-1, \hat{k} \leq \hat{s} \leq s-k+\dot{k}\right\} \qquad \text{for } K(j) \geq k \geq 2.\\ \mathbf{S}(k,s) &:= \left\{E(k,\acute{s}); E(\hat{k},\hat{s}) :\ \forall 1 \leq \acute{s} \leq s-1, 1 \leq \hat{k} \leq k-1, \mathbf{K}(k,s)\right\}.\end{aligned} \tag{7.22}$$

The event $E(k,s)$ encompasses the sample paths in which the number of class

(j,k) jobs completed up till the end of cycle s, is at least equal to that in the JS2 operating under the restricted policy π^*_ϵ. For the first job class, i.e., job class $(j,1)$ we readily obtain the following expression for the probability that $E(1,s)$ occurs:

$$\mathbf{P}\left(\bar{X}^\omega(j,1,sC) \geq X^\epsilon\omega(j,1,sC)\right) \geq \{\mathbf{P}\left(\bar{r}^\omega_1(j,1) \geq r^\epsilon_1\omega(j,1)\right)\}^s$$
$$\Rightarrow \mathbf{P}(E(1,2)) \geq \left(1 - e^{-\ddot{\check{\gamma}}^\omega(\epsilon)\ddot{f}^\omega(\epsilon)}\right)^{-M(\rho(j,1))s}, \tag{7.23}$$

where $\ddot{\check{\gamma}}^\omega(\epsilon) = \min_{(j,k)\in\mathbf{H}} \{\check{\gamma}^\omega(j,k,\epsilon)\}$ and $\ddot{f}^\omega(\epsilon) = \min_{(j,k)\in\mathbf{H}} \{f^\omega(j,k,\epsilon)\}$. In the following lemma we establish a similar probabilistic bound for relating to $E(k,s)$.

Lemma 20. *For a given $\epsilon \in (0,1)$, $j \in \Phi$ and $\omega \in \Omega$, and any cycle $s \geq k, K(j) \geq k \geq 2$, under policy π^*_ϵ the following holds:*

$$\boldsymbol{P}_\omega\left(E(k,s) \mid E(k,s-1), E(k-1,s-1)\right) \geq \left(1 - e^{-\ddot{\gamma}^\omega(\epsilon)\ddot{f}(\omega,\epsilon)}\right)^{M(\rho(j,k))}. \tag{7.24}$$

Proof. In the JS2 operating under π^*_ϵ, for the number of jobs completed for a particular class (j,k), we have

$$X^\epsilon(j,k,sC) \leq X^\epsilon(j,k,s-1) + M(\rho(j,k))\check{\gamma}^\omega(j,k,\epsilon) \tag{7.25a}$$
$$X^\epsilon(j,k,sC) \leq X^\epsilon(j,k,s-1) + a^\omega(j,k). \tag{7.25b}$$

Define event $\grave{E}(k,s) := \left\{\bar{Q}^\omega_s(j,k) \geq \check{\gamma}^\omega(j,k)M(\rho(j,k))\right\}$ for cycle $s \geq k$. We are interested if this condition holds at the beginning of a cycle s. If $\grave{E}(k,s)$ holds then $\bar{r}^\omega_s(j,k)$ satisfies (7.20) and from (7.25a) and (7.4b) we have the following relation:

$$\begin{aligned}&\mathbf{P}_\omega\left(\bar{X}^\omega(j,k,sC) \geq X^\epsilon\omega(j,k) \mid E(k,s-1), E(k-1,s-1)\right)\\ &\geq \mathbf{P}_\omega\left(\bar{r}^\omega_s(j,k) \geq M(\rho(j,k))\check{\gamma}^\omega(j,k,\epsilon)\right) \geq \left(1-e^{-\ddot{\gamma}^\omega(\epsilon)\ddot{f}(\omega,\epsilon)}\right)^{M(\rho(j,k))}. \quad (7.26)\end{aligned}$$

If $\grave{E}(k,s)$ does not hold then from (7.25b) and (7.4a), we have

$$\begin{aligned}&a^\omega(j,k) + \bar{X}^\omega(j,k,\acute{s}C) - \bar{X}^\omega(j,k-1,\acute{s}C) = \bar{Q}^\omega_s(j,k)\\ \Rightarrow\ &\mathbf{P}\left(\bar{X}^\omega(j,k,sC) = a^\omega(j,k) + \bar{X}^\omega(j,k-1,\acute{s}C)\right) = \mathbf{P}\left(\bar{r}^\omega_s(j,k) = \bar{Q}^\omega_s(j,k)\right)\\ &\Rightarrow \mathbf{P}\left(E(k,s) \mid E(k-1,\acute{s})\right) \geq \left(1-e^{-\ddot{\gamma}^\omega(\epsilon)\ddot{f}(\omega,\epsilon)}\right)^{M(\rho(j,k))},\end{aligned} \quad (7.27)$$

where $\acute{s} = s-1$. From expressions (7.26) and (7.27), the result follows. □

In the form of (7.24), we have a bound on the probability of $E(k,s)$, conditioned on occurrence of $E(k-1,s-1)$ and $E(k,s-1)$. Taking advantage of this fact, we try to compute a bound on the probability of $E(k,s)$ by successive conditioning as shown below,

$$\begin{aligned}\mathbf{P}_\omega(E(k,s)) &\geq \mathbf{P}_\omega(E(k,s) \mid \mathbf{S}(k,s))\mathbf{P}_\omega\left(\mathbf{S}(k,s)\right)\\ &= \mathbf{P}_\omega\left(E(k,s) \mid E(k-1,s-1), E(k,s-1)\right)\mathbf{P}_\omega\left(\mathbf{S}(k,s)\right)\\ &= \prod_{(\hat{k},\hat{s})\in\mathbf{K}(k,s)}\left(\mathbf{P}_\omega\left(E(\hat{k},\hat{s}) \mid E(\hat{k}-1,\hat{s}), E(\hat{k}-1,\hat{s}-1)\right)\right).\end{aligned} \quad (7.28)$$

This result gives a bound on the probability of $E(k,s)$ and a probabilistic bound on the time required to drain the job type. Generalized to all job types we obtain the result below. Note that $\mathfrak{c}^\omega(j,k,\epsilon)$ is number of cycles required to complete all the jobs of class (j,k) under scenario ω, when policy π^*_ϵ is followed.

Lemma 21. *If policy π^*_{fi} is followed in a GJS with doubly stochastic processing times and*

$$C_f \frac{\boldsymbol{u}(j,k)}{m^\omega(j,k)} \geq 1 + \frac{1}{1-\epsilon} \quad \forall \quad (j,k) \in \boldsymbol{H}, \tag{7.29}$$

for some $\omega \in \Omega$ and $1 > \epsilon > 0$, then

$$\boldsymbol{P}_\omega \left(\frac{\bar{T}^\omega}{\boldsymbol{T}^\omega} \leq \dot{ub}^\omega \right) \geq \left(1 - e^{-\ddot{\check{\gamma}}^\omega(\epsilon)\ddot{f}(\omega,\check{\epsilon})}\right)^{J\dot{M}\dot{c}^\epsilon(\omega)\dot{K}}, \tag{7.30}$$

where

$$\begin{aligned}
\dot{M} &:= \max_{j\in\Phi} M(j), \quad \dot{K} := \max_{j\in\Phi} K(j) \\
\dot{\check{\gamma}}^\omega(\epsilon) &:= \max_{(j,k)\in\boldsymbol{H}} \check{\gamma}^\omega(j,k,\epsilon), \quad \dot{\mathfrak{c}}^\omega(\epsilon) = \max_{(j,k)\in\boldsymbol{H}} \mathfrak{c}^\omega(j,k,\epsilon) \\
\dot{ub}^\omega &= \max_{(j,k)\in\boldsymbol{H}} \left(\frac{C_f\boldsymbol{u}(j,k)}{(1-\epsilon)C_f\boldsymbol{u}(j,k) - (2-\epsilon)m^\omega(j)} \right) + \frac{C_f}{\boldsymbol{T}^\omega}.
\end{aligned}$$

Proof. Considering (7.28) for all end job classes $(j,K(j))$, we obtain

$$\begin{aligned}
&\left(1 - e^{-\ddot{\check{\gamma}}^\omega(\epsilon)\ddot{f}(\omega,\check{\epsilon})}\right)^{\sum_{\acute{k}\leq k}\left(M(\rho(j,\acute{k}))(\mathfrak{c}^\omega(j,k,\epsilon)-\acute{k})\right)} \leq \mathbf{P}_\omega(\bar{\mathfrak{c}}^\omega(j,k) \geq \mathfrak{c}^\omega(j,k,\epsilon)) \\
&\qquad \Rightarrow \left(1 - e^{-\ddot{\check{\gamma}}^\omega(\epsilon)\ddot{f}(\omega,\check{\epsilon})}\right)^{J\dot{M}\dot{\mathfrak{c}}^\omega(\epsilon)\dot{K}} \leq \mathbf{P}_\omega\left(\bar{T}(\omega) \leq \max_{j\in\Phi} \mathfrak{c}^\omega(j,K(j),\epsilon)C\right).
\end{aligned} \tag{7.31}$$

Also,

$$
\begin{aligned}
T^{\epsilon}(\omega) &= \max_{j\in\Phi} \mathfrak{c}^{\epsilon}\omega(j,K(j))C \\
&\le C \max_{(j,k)\in\mathbf{H}} \left(\left\lceil \frac{b^{\omega}(j,k)}{M(\rho(j,k))\check{\gamma}^{\omega}(j,k,\epsilon)} \right\rceil + K(j) - 1 \right) \\
\Rightarrow \frac{T^{\omega}(\epsilon)}{\mathbf{T}^{\omega}} &\le \dot{u}\dot{b}^{\omega} + \frac{\dot{K}C}{\mathbf{T}^{\omega}}, \qquad (7.32)
\end{aligned}
$$

where the final inequality is obtained by modifying (6.36) and (6.33) for the general job shop case. □

We have derived a probabilistic upper bound on the makespan under policy π_f^* and hence under policy π_{fi}^*, since π_f^* performs better than π_{fi}^*. Next we try to explore a lower bound. For this purpose we look at the CJS, the continuous relaxation of the generic job shop. When the processing time are random, combining relations (5.12e)–(5.12g), we obtain the following, specifically when the fluid allocation is $\mathbf{u}$:

$$
\begin{aligned}
&\bar{\bar{X}}^{\omega}(j,k,\bar{\bar{t}}(j,\omega)) = b^{\omega}(j,k) \\
\Rightarrow &\sum_{i=1} b^{\omega}(j,k)\bar{m}_i^{\omega}(j,k) \le \bar{\bar{t}}(j,\omega)\mathbf{u}(j,k). \qquad (7.33a)
\end{aligned}
$$

As argued while deriving (6.38), we observe that for the generic job shop due to (7.33a), the bounds (6.18) and hence (6.17) holds in modified form, i.e.,

$$
\mathbf{P}_{\omega}\left(\bar{T}(\omega) \ge \mathbf{T}(\omega)\left(1 - \frac{\epsilon}{\ddot{m}(\omega)}\right)\right) \ge \left(1 - e^{-\ddot{a}(\omega)\ddot{f}^{\omega}(\epsilon)}\right)^{J\dot{K}}. \qquad (7.34)
$$

Combining (7.34) and Theorem 21 we get a probabilistic bound, which indicates that for large number of jobs, with high probability (ϵ-dependent) the makespan under a given scenario $\omega \in \Omega$ lies within tight bounds around the associated optimal fluid makespan.

7.2.1 Asymptotically Optimal

In this subsection we prove that suggested heuristic π_f^* is asymptotically optimal. For this purpose, by modifying restrictions on π_{fi}^* we construct a policy $\tilde{\pi}_{fi}^*$. A job initially in job class (j,k) at time $t = t_0$ is said to be labeled (j,k).

Definition 10. *The parameters $(\boldsymbol{z}_{fi}, \boldsymbol{C}_{fi})$ of policy $\tilde{\pi}_{fi}^*$ are same as that of policy π_{fi}^* given in Definition 8. Further,*

a. *Jobs labeled (j,k) which were processed in previous cycle at step $k' \geq k$ are considered eligible for processing in present cycle at step $k' + 1 \geq K(j)$.*

b. *Jobs arriving at a station in the present cycle which are not processed are returned to the previous step.*

c. *In scenario $\omega >$, if $b^\omega(j,k) > 0$, the maximum time spent in a cycle processing jobs labeled (j,k') at step $k' \leq k$, is $z_{fi}^\omega(j,k,k') := C_{fi}\boldsymbol{u}(j,k)\frac{a^\omega(j,k')}{b^\omega(j,k)}$.*

d. *In scenario ω at server $l \in \varpi(j,k)$, the maximum number of (j,k') labeled jobs, $k' \leq k$, processed in a cycle is $\varrho^\omega(j,k,k',l) := \left\lfloor \frac{z_{fi}^\omega(j,k,k')}{m^\omega(j,k)} \right\rfloor$.*

The above condition (**b**) is similar to some manufacturing settings, where there is time window within which the next step has to be completed. In out model, the 'returned job' might not physically repeat the process but is assigned a processing time. This modeling modification is for ease of analysis.

Assuming jobs are always available, the number of jobs labeled (j,k') that can be processed at step $k \geq k'$ in scenario ω at server $l \in \varpi(j,k)$ is

$\bar{\varrho}^{\omega}(j,k,k^{'},l)$. The total number of jobs labeled (j,k) processed in a cycle at the station $\rho(j,k)$ is $\bar{\bar{\varrho}}^{\omega}(j,k,k^{'})$. Based on these definition and the problem restriction, we observe that after $K(j)$ cycles the number of jobs label (j,k) that are completed in a cycle can be given by the following relation:

$$\ddot{\varrho}^{\omega}(j,k) := \min_{k^{'}\geq k}\left\{\bar{\bar{\varrho}}^{\omega}(j,k,k^{'})\right\}. \tag{7.35}$$

The above definition is based on the condition that jobs are always available at job class (j,k). As in the discussion preceding equations (6.39a) and (6.39b), for each scenario ω and job class (j,k) consider a sequence of iid random variables $\{\bar{m}_1^{\omega}(j,k),\bar{m}_2^{\omega}(j,k),\ldots\}$. The number of cycle required after $K(j)$ cycles to complete jobs labeled (j,k) is $\bar{\varsigma}^{\omega}(j,k)$. Similar to (6.39a) and (6.39b), we have the following relations:

$$\bar{\varrho}_1^{\omega}(j,k,k^{'},l) = \min\left\{\varrho^{\omega}(j,k,k^{'}), \min\left\{s : \sum_{i=1}^{s+1}\bar{m}_i^{\omega}(j,k) > z_{fi}^{\omega}(j,k,k^{'})\right\}\right\} \tag{7.36}$$

$$\bar{\varsigma}^{\omega}(j,k) = \min\left\{s : \sum_{i=1}^{s+1}\ddot{\varrho}_i^{\omega}(j,k) \geq a^{\omega}(j,k)\right\}. \tag{7.37}$$

Note that $\bar{\varsigma}^{\omega}(j,k)$ is a stopping time for the sequence of random variables $\{\ddot{\varrho}_1^{\omega}(j,k),\ddot{\varrho}_2^{\omega}(j,k),\ldots\}$. Applying *Wald's equation* to (7.37) and using the fact $K(j)-k$ cycles elapse before the job labeled (j,k) is actually processed at the last step, we get the following result.

Lemma 22. *If policy $\tilde{\pi}_{fi}^{*}$ is followed, under scenario $\omega \in \Omega$, the expected* *makespan of job class (j,k) satisfies*

$$\boldsymbol{E}_\omega\left[\bar{t}^\omega(j,k)\right] \leq \frac{a^\omega(j,k)C_f}{\boldsymbol{E}_\omega\left[\ddot{\varrho}^\omega(j,k)\right]} + (K(j)-k+1)C_f. \tag{7.38}$$

To study the asymptotic behavior, consider a sequence of scaled job shops following fluid or HW scaling. $\mathbf{J}_n$ is the n^{th} job shop in the sequence. In analysis that follows a subscript n indicated the scaling. Under these scalings we obtain an expression for the number of labeled jobs completed in a cycle at each step. Recall that in (6.29)–(6.30c) we define two real valued functions $g(n)$ and $f(n)$.

Lemma 23. *If policy $\tilde{\pi}^*_{fi}$ is followed, C^n_{fi} satisfies condition (6.43) then*

$$\lim_{n\to\infty}\left(\boldsymbol{E}_\omega\left[\frac{\bar{\varrho}^\omega_n(j,k,k^{'},l)}{C^n_{fi}}\right]\right) = \boldsymbol{u}(j,k)\frac{a^\omega(j,k^{'})}{b^\omega(j,k)}.$$

Proof. From definition of $\bar{\varrho}^\omega_n(j,k,k^{'},l)$ we have,

$$\mathbf{E}_\omega\left[\bar{\varrho}^\omega_n(j,k,k^{'},l)\right] = \varrho^\omega_n(j,k,k^{'},l)\mathbf{P}_\omega\left(\sum_{i=1}^{\varrho^\omega_n(j,k,k^{'},l)-1}\bar{m}^\omega_i(j,k)\leq z^\omega_{fi}(j,k,k^{'})\right) + \sum_{s=1}^{\varrho^\omega_n(j,k,k^{'},l)-1} s\mathbf{P}_\omega\left(\sum_{i=1}^{s+1}\bar{m}^\omega_i(j,k)\geq z^\omega_{fi}(j,k,k^{'});\sum_{i=1}^{s}\bar{m}^\omega_i(j,k)\leq z^\omega_{fi}(j,k,k^{'})\right)$$

Considering scaling and taking limits on both side of the equation above we obtain

$$\Rightarrow \lim_{n\to\infty}\left(\mathbf{E}_\omega\left[\frac{\bar{\varrho}_n^\omega(j,k,k^{'},l)}{C_{fi}^n}\right]\right) = \lim_{n\to\infty}\left(\frac{\varrho_n^\omega(j,k,k^{'},l)}{C_{fi}^n}\right)$$

$$= \lim_{n\to\infty}\left\lfloor\frac{z_{fi}^\omega(j,k,k^{'})a^\omega(j,k)}{b^\omega(j,k)m^\omega(j,k)C_{fi}^n}\right\rfloor.$$

The first equality holds because of convergence in probability. □

Lemma 24. *If conditions in Lemma 23 are satisfied, then for each $\omega \in \Omega$,*

$$\lim_{n\to\infty}\frac{T_n^\omega}{n\,\boldsymbol{T}^\omega} = 1.$$

Proof. Now applying Lemma 10 along with Lemma 23 to the definition of $\ddot{\varrho}^\omega(j,k)$, we have

$$\lim_{n\to\infty}\left(\mathbf{E}_\omega\left[\frac{\ddot{\varrho}_n^\omega(j,k)}{\dot{M}_n C_f^n}\right]\right) = \min_{k^{'}\geq k}\left(\frac{\mathbf{u}(j,k^{'})a^\omega(j,k)M(\rho(j,k^{'}))}{\dot{M}b^\omega(j,k^{'})}\right), \tag{7.39}$$

where $\dot{M} = \max i \in \Psi M(i)$. Combining (7.39) and Lemma 10, we get

$$\lim_{n\to\infty}\left(\mathbf{E}_\omega\left[\frac{\bar{t}_n^\omega(j,k)}{n}\right]\right) = \lim_{n\to\infty}\left(\frac{a_n^\omega(j,k)}{n\dot{M}_n}\right)\frac{1}{\lim_{n\to\infty}\mathbf{E}_\omega\left[\frac{\ddot{\varrho}_n^\omega(j,k)}{\dot{M}_n C_f^n}\right]}$$

$$= \frac{a^\omega(j,k)}{\dot{M}}\max_{k^{'}\geq k}\left\{\frac{b^\omega(j,k^{'})\dot{M}}{a^\omega(j,k)M(\rho(j,k^{'}))\mathbf{u}(j,k^{'})}\right\}$$

$$= \mathbf{t}^\omega(j,k)$$

$$\Rightarrow \lim_{n\to\infty}\left(\frac{\bar{T}_n^\omega}{n}\right) = \mathbf{T}^\omega$$ □

From the lemma above, the asymptotic optimality of the policy $\tilde{\pi}_{fi}^*$ and hence that of π^*_f directly follows. Note that though limit considered is guaranteed to exist under $\tilde{\pi}^*_{fi}$, the existence of limited under π^*_f is not immediately evident. Considering the lower bound as argued in Theorem 15, this issue can be resolved. Hence, in the below theorem we safely assume that the limit exists.

Theorem 19. *If policy π^*_f is followed and C^n_f satisfies the following:*

$$\lim_{n\to\infty} \frac{C^n_f}{n} = 0 \quad \textit{and} \quad \lim_{n\to\infty} \frac{1}{C^n_f} = 0,$$

then

$$\lim_{n\to\infty} \frac{T_n(\pi^*_f)}{n} = \boldsymbol{T}.$$

7.3 Conclusions

In this dissertation we have developed cyclic policies for scheduling the stochastic job shop. These policies are constructed based on certain known quantities of the job shop and an optimal solution of the associated fluid model. The suggested cyclic policies were shown to be asymptotically optimal under appropriate scaling. Also, within the setting of doubly stochastic processing times the makespan of such policies are close to the optimal makespan with high probability, for each scenario.

In generic job shop scheduling problems as the number of jobs increases, the combinatorial complexity is a major concern. The fluid model we use for construction of the suggested policies ignores such combinatorial details of the problem. So, in the makespan problem considered as the number of jobs increases, the combinatorial structure does not affect the construction of the policy and in fact our approximation scheme becomes more effective.

Due to the aforementioned advantage of the irrelevance of the combinatorial details of the problem, fluid models have been used in the literature to construct policies which are asymptotically optimal. We extended this methodology to include parameter uncertainty.

Scheduling a job shop can be a difficult problem depending on the setting. The fluid following policies developed in the literature to solve the makespan problem, revolve around the notion that idleness at the bottleneck station should be minimized. Determining the bottleneck depends upon knowledge of the system workload. Hence, when parameters are uncertain as in our setting, the controller might not to be able to directly determine the bottleneck. This changes the nature of the problem. If a class of all non-anticipatory, scenario independent policies is considered fluid model is just continuous relaxation of the discrete job shop. The optimization problem is a separated continuous time linear program which itself might be difficult to solve.

Restricting to the class of cyclic policies greatly reduces the complexity of the associated fluid problem. However, it is not immediately clear that the associated fluid model will give an lower bound on the achievable expected makespan in case of general job shop operating (GJS)under cyclic constraints (CYC). This question have been answered in this dissertation. Based on the result that fluid model does provide a lower bound, we proved asymptotic optimality of the suggested policy.

An interesting problem to explore is the setting under which the controller is able to recognize the bottleneck at time $t = t_0$ under each scenario but the policy is fixed a priori as in this dissertation. Another interesting variation would be to explore the set of data or information like that is sufficient for effective scheduling.

A significant area for further investigation is to explore other cost functionals. In particular, the weighted holding cost objective has been of great interest in the scheduling and queueing communities. Based on literature, we suspect that when the class of non-anticipatory, scenario independent policies is considered, an asymptotically optimal fluid following policy can be constructed for the objective of weighted holding cost. However, when we restrict to cyclic policies and such objective functions, it is not clear if the associated fluid model provides a lower bound. Answering this question would be key step to extend the present methodology to more general cost functions in the stochastic job shop setting.

Bibliography

[1] B. Ata. Dynamic power control of a wireless static channel subject to a quality-of-service constraint. *Operational Research*, 53(5):842–851, 2005.

[2] B. Ata and S. Shneorson. Dynamic control of an $M/M/1$ service system with adjustable arrival and service rates. *Management Science*, 52(11):1778–1791, 2006.

[3] D. P. Bertsekas. *Dynamic programming and optimal control*, volume 2. Athena Scientific, 2001.

[4] D. Bertsimas and D. Gamarnik. Asymptotically optimal algorithms for job shop scheduling and packet routing. *Journal of Algorithms*, 33(2):296–318, 2003.

[5] D. Bertsimas, D. Gamarnik, and J. Sethuraman. From fluid relaxations to practical algorithms for high-multiplicity job-shop scheduling: the holding cost objective. *Operations Research*, 51(5):798–813, 2003.

[6] D. Bertsimas and J. Sethuraman. From fluid relaxations to practical algorithms for job shop scheduling: the makespan objective. *Mathematical Programming*, 1:61–102, 2002.

[7] R. A. Bowman. Job release control using a cyclic schedule. *Production and Operations Management*, 11(2):247–287, 2002.

[8] B. Büke, J. J. Hasenbein, and D. P. Morton. Minimizing makespan in a multiclass fluid network with parameter uncertainty. 2007. Submitted.

[9] H. Chen and D. Yao. Dynamic scheduling of a multiclass fluid network. *Operations Research*, 41:1104–1115, 1993.

[10] D. Connors, G. Feigin, and D. Yao. Scheduling semiconducotor lines using a fluid network model. *IEEE Transactions on Robotics and Automation*, 10:88–98, 1994.

[11] J. G. Dai and G. Weiss. Stability and instability of fluid models for reentrant lines. *Mathematics of Operations Research*, 21:115–134, 1996.

[12] J. G. Dai and G. Weiss. A fluid heuristic for minimizing makespan in job shops. *Operations Research*, 50(4):692–707, 2002.

[13] J. M. George and J. M. Harrison. Dynamic control of a queue with adjustable service rate. *Operational Research*, 49(5):720–731, 2001.

[14] D. Gross and C. M. Harris. *Fundamentals of Queueing Theory*. Wiley, 3rd edition, 1998.

[15] X. Guo and O. Hernández-Lerma. Drift and monotonicity conditions for continuous-time controlled markov chains with an average criterion. *IEEE Transactions on Automatic Control*, 48(2):236–245, 2003.

[16] S. Halfin and W. Whitt. Heavy-traffic limits for queues with many exponential servers. *Operations Research*, 29:567–588, 1981.

[17] N. G. Hall, T.-E. Lee, and M. E. Posner. The complexity of cyclic job shop scheduling problems. *Journal of Scheduling*, 5(4):307–327, 2002.

[18] O. Korbaa, H. Camus, and J. Gentina. A new cyclic scheduling algorithm for flexible manufacturing systems. *The International Journal of Flexible Manufacturing Systems*, 14(2):173–187, April 2002.

[19] P. R. Kumar. Re-entrant lines. *Queueing Systems: Theory and Applications*, 13:87–110, 1993.

[20] S. H. Lu and P. R. Kumar. Distributed scheduling based on due dates and buffer priorities. *IEEE Transactions on Automatic Control*, 36:1406–1416, 1991.

[21] C. Maglaras. Discrete-review policies for scheduling stochastic networks: trajectory tracking and fluid-scale asymptotic optimality. *The Annals of Applied Probability*, 10(3):897–929, 2000.

[22] C. Maglaras. Continuous-review tracking policies for dynamic control of stochastic networks. *Queueing Systems*, 43(1–2):43–80, January 2003.

[23] M. L. Puterman. *Markov decision processes.* Wiley, New York, 1994.

[24] S. M. Ross. *Stochastic processes.* Wiley, New York, 2nd edition, 1996.

[25] R. Roundy. Cyclic schedules for job-shops with identical jobs. *Mathematics of Operations Research*, 17(4):842–865, April 1992.

[26] H. L. Royden. *Real analysis.* MacMillan, New York, 3rd edition, 1988.

[27] L. I. Sennott. *Stochastic dynamic programming and the control of queueing systems.* Wiley, New York, 1999.

[28] M. Y. Wang, S. P. Sethi, and S. L. V. D. Velde. Minimizing makespan in a class of reentrant shops. *Operations Research*, 45(4):702–712, Sep–Oct 1997.

[29] G. Weiss. On optimal draining of fluid reentrant lines. In F. P. Kelly and R. J. Williams, editors, *Stochastic Networks*, volume 71 of *The IMA volumes in mathematics and its applications*, pages 93–105, New York, 1995. Springer-Verlag.

[30] J. Wijngaard and S. Stidham. Forward recursion for Markov decision processes with skip-free-to-the-right transitions, part I: theory and algorithm. *Mathematics of Operational Research*, 11:295–308, 1986.

Vita

Kranthi Mitra Adusumilli was born in Gudivada, India on December 22, 1979, the son of Adusumilli Venkata Subba Rao and Adusumilli Sandhya Sri. He received his Bachelor of Technology degree in 2002 in Production and Industrial Engineering from Indian Institute of Technology, New Delhi, India. After his undergraduate studies he joined graduate program in Operations Research and Industrial Engineering at University of Texas at Austin where in 2005, he earned the Master of Science degree.

Permanent Address: 1-43/2 Shankarnagar
Hyderabad, AP-500050
India

This dissertation was typeset with LaTeX 2_ε[1] by the author.

[1] LaTeX 2_ε is an extension of LaTeX. LaTeX is a collection of macros for TeX. TeX is a trademark of the American Mathematical Society. The macros used in formatting this dissertation were written by Dinesh Das, Department of Computer Sciences, The University of Texas at Austin, and extended by Bert Kay, James A. Bednar, and Ayman El-Khashab.